

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

P-89

LANGLEY
GRANT
IN-6/CR
93190

TITLE

Analysis of Adaptive Algorithms
for an Integrated Communication Network

First
Semiannual Status Report
June 1, 1985 - November 30, 1985

NASA Grant No. NAG-1-595

Project Personnel

Graduate Research Assistants

Matthew Barr
Chong-kwon Kim

Principal Investigator

Daniel A. Reed

Daniel A. Reed

(NASA-CR-181263) ANALYSIS OF ADAPTIVE
ALGORITHMS FOR AN INTEGRATED COMMUNICATION
NETWORK Semiannual Status Report, 1 Jun. -
30 Nov. 1985 (Illinois Univ.) 89 p Avail:
NTIS HC A05/MF A01

N87-28298

CSC 09B G3/61

Unclass
0093190

OVERVIEW

The first six months of the project saw the employment of Matthew Barr and Chong-kwon Kim as research assistants. Although the grant period began in June, the commitment to support the research arrived too late to employ students for the summer. Hence, both Barr and Kim began work in the fall. Each has been working on slightly different, but complementary, approaches to the support of heterogeneous traffic (i.e., voice, video, and data) on a single, integrated network.

Barr has begun examining techniques that trade communication bandwidth for decreased transmission delays. When the network is lightly used, these schemes attempt to use additional network resources to decrease communication delays. As the network utilization rises, the schemes degrade gracefully, still providing service but with minimal use of the network. Because the schemes use a combination of circuit and packet switching, they should respond to variations in the types *and* amounts of network traffic. Because Barr just began this work in September, there are few results to report thus far. As the work progresses, it will be discussed in more detail in future reports.

Drawing on Reed's earlier work on integrated communication networks, Kim had already begun (in the spring) to examine some of the design issues. Hence, when the fall semester began, he was able to quickly begin work. The approach we are currently investigating uses a combination of circuit and packet switching to support the widely varying traffic demands imposed on an integrated network. The packet switched component is best suited to bursty traffic where some delays in delivery are acceptable. The circuit switched component is reserved for traffic that must meet real-time constraints.

With these goals in mind, we began with a detailed examination of the prior work in the area, concentrating on primarily on two issues:

- implementation strategies and
- performance studies.

Following that survey, we began simulating selected packet routing algorithms that might be used in an integrated network. Although it may seem that studying packet routing in isolation is of limited value, an integrated traffic places widely varying workload demands on a network. One of the key goals of our research is identification of *adaptive* algorithms, ones that can respond to both the transient and evolutionary changes that arise in integrated networks. Hence, results from this packet switching study are directly applicable to integrated networks. Based on this work, we have developed a new algorithm, hybrid weighted routing, that adapts more effectively to changes in the presented workload than any previously known.

We are currently simulating hybrid weighted routing, and other algorithms, in an *integrated* network. In addition, we are extending our simulation studies to include more realistic network conditions. These goals of our current research are discussed in more detail in the final section of this report.

TABLE OF CONTENTS

SECTION

1. INTRODUCTION	1
1.1. Heterogeneous Traffic	1
1.2. Network Design Issues	4
1.3. Switching Methods	6
1.4. The Integrated Network	11
1.5. Overview of Following Sections	14
2. PRIOR WORK	16
2.1. Packet Switching	16
2.2. Circuit Switching	31
2.3. Hybrid Network	33
3. MOTIVATION AND METHODOLOGY OF PACKET ROUTING PERFOR- MANCE STUDY	46
3.1. Introduction	46
3.2. Previous Performance Studies	47
3.3. Hybrid Weighted Routing	49
3.4. Motivation of Packet Routing Performance Study	51
3.5. Methodology	52
4. PERFORMANCE OF PACKET ROUTING ALGORITHMS	58
4.1. Introduction	58
4.2. Simulation Traffic	58
4.3. Simulation Results	63
4.4. Summary	75
5. FUTURE WORK	78
REFERENCES	82

1. INTRODUCTION

1.1. Heterogeneous Traffic

Conventional communication networks were designed to provide human voice transmissions. Telephone networks assign 4 KHz analog end-to-end connections to each voice call. A voice call uses an assigned connection exclusively until it finishes its conversation. The 4 KHz circuit is sufficient because of the frequency limitation of human voice. In voice communications, end users are human beings, and the end users perform intellectual functions such as voice generation, voice interpretation, and transmission error detection and correction. In these senses, the low bandwidth, error-prone analog 4 KHz channel well satisfies the requirements of voice communications.

More recently, communication systems have been forced to support data transmission between computers and end users. Data traffic consists of at least two classes: short, interactive data traffic that requires prompt delivery and bulk data traffic. Interactive, query/response, and database updates belong to the first class, while a file transfer belongs to the latter class. In addition to data traffic, future communication systems must support increasing video communication traffic demands. Video traffic includes still image transfers (e.g. facsimiles) and near-real-time moving image communications (e.g. picture phones and videotex). Voice, data, and video traffic each have their own distinctive traffic characteristics. The existence of such heterogeneous traffic (voice, data, and video) in one network system creates new network design problems that cannot be satisfactorily solved by a conventional homogeneous network design approach. A brute force solution, just increasing the capacity of existing telephone or data networks to support other types of traffic, is not ideal

because a network designed for homogeneous traffic cannot handle other classes of traffic effectively. An optimal system must have mechanisms to efficiently satisfy the distinctive characteristics of all traffic types.

Table 1.1 compares the characteristics and requirements of various types of traffic. Voice traffic must be synchronously transmitted. In a synchronized transmission, the receiving data rate and the arrival order of data are same as the sending data rate and the departure order of the data. Voice traffic demands a near-realtime delivery (preferably, delay \leq 500 msec). A slow voice transmission would adversely affect the smoothness of conversations.

Table 1.1. Characteristics of voice, video, and data traffic.

Characteristics	Traffic classes		
	Voice	Video	Data
Synchronized	Yes	Yes (moving image)	No
Delay	< 500 msec	< 500 msec (moving image)	< 1 sec (interactive) Days (bulk)
Error control		Generally Yes	Yes
Activity ratio	High	High	Low (interactive)
Transmission rate	Small	Large	Small (interactive)
Session length	Short		Long
Access method	Blocking	Blocking (moving image)	Delay
Preferred switching	Circuit	Circuit	Packet (interactive)

Voice traffic has short *session duration time* (a cycle of connection setup, data transfer, and disconnection is called a session) compared to an interactive terminal session, however the *activity ratio* (a session is said to be active when meaningful signals are transmitted) of voice traffic is generally much higher than that of interactive data traffic. Telephone communication systems use some form of circuit switching to meet the fast, synchronous transmission requirements of voice traffic. As mentioned earlier, telephone systems handle voice traffic with low bandwidth channels. Depending on the image quality, real-time video traffic can require a bandwidth of up to one million bit/sec. Interactive data traffic is characterized by a long conversation session with low activity ratio, because it consists of typical think-type-response cycles. Data traffic does not require synchronous transmissions, but its delays must be short and reasonably consistent (usually less than one second). In contrast, bulk data traffic does not require immediate transmissions.

Most existing communication networks experience wide variations in traffic demand. The traffic variations can be either transient or evolutionary. The transient traffic fluctuations occur during relatively short time periods (e.g. during a day or a week). For example, the peak voice traffic demand during a day in a telephone communication system is much higher than the average traffic load. In addition to the transient traffic variations, network systems have slowly changing, evolutionary traffic load variations. The long-term traffic load variation appears as a traffic demand trend over a long time period.

Heterogeneous traffic shows more complex patterns of workload variations. First, each traffic class has its own transient and long-term workload changes. Second, the sum of all traffic workloads has similar workload variation patterns as individual traffic. Third and most importantly, the traffic mix (proportion of each traffic workload to the total workload)

changes dynamically. The traffic mix change can be transient; for example, one type of traffic (say data traffic) might dominate on weekdays and another type (say voice traffic) might dominate on weekends. In addition to the transient traffic mix, development of new technologies and introduction of new applications cause evolutionary variations of the traffic mix. For example, wide acceptance of distributed systems, database systems, and office information systems will increase data traffic demands.

1.2. Network Design Issues

Cost Effectiveness: One of the most important network design issues is the cost effectiveness of communication systems. The problem is finding the most economical communication network that satisfies specified throughput (workload) and delay requirements. Frequently, the problem includes other constraints such as network topology constraints (e.g. number of nodes and degree of connections). The problem is so complex that no closed-form, analytic solutions have been developed for the problem. All proposed solutions [FrCh72], [BoFr77] had an iterative form that consisted of initial topology generation, traffic flow and trunk capacity assignment, and network modification steps. First, in the initial topology generation step, a crude network topology is initially generated. For this topology, the flow and capacity assignment procedure tries to optimally assign traffic flows and transmission capacity to each link. The network modification procedure then makes small topological changes to the initial topology. Traffic flows and link capacity are reassigned optimally to the modified topology. If the modified network is better than the original network, the original network is replaced with the modified network. The network modification and flow/capacity assignment steps are then repeated until a satisfactory result is obtained.

In the network design process described above, network designers consider workload fluctuations and try to guarantee specified network performance even during the periods of peak workloads. Usually, excess transmission capacity is provided for the peak workloads. The total excess capacities for the peak workloads would be enormous in a heterogeneous network (a network with heterogeneous traffic) if the network provides separate excess capacity for each traffic class.

Adaptability: Another network design issue is the adaptability of network systems. If the network conditions do not change, we do not need to adjust network control parameters dynamically. For example, it is not necessary to change optimal fixed routing paths that were determined at network initiation. However, routing paths must be readjusted to maintain optimality when the network condition fluctuates. A network's condition is defined not only by the states of traffic workloads but also by the states of network components (nodes and trunks). Although traffic workloads change more dynamically than network components, changes in network components (e.g. transmission line failure/recoveries) have greater effects on network performance. An adaptive network senses network status fluctuations and adapts its operations to manage the fluctuations. For example, an adaptive routing algorithm avoids temporarily congested paths and sends packets along idle or lightly loaded paths. Telephone companies takes advantage of different time zones (with different peak traffic periods) in servicing long distance calls. Thus, good adaptive routing algorithms enhance network performance under dynamic network status changes.

Flexibility: A unique requirement of a heterogeneous network is flexibility. We pointed out earlier that the traffic mix of heterogeneous traffic changes dynamically. Suppose a heterogeneous network adopts the following design approach: network resources are parti-

tioned according to traffic class and are not sharable. Assume a situation where most of the voice traffic resources are idle because of low voice traffic while data traffic resources are overloaded with heavy data traffic. The data traffic cannot use voice traffic resources even though the resources are not used by the voice traffic. In addition to the transient traffic mix variations, this approach cannot adjust conveniently to long-term traffic mix changes. Manual intervention is needed to reassign resources to each traffic class in case of the long-term traffic mix changes. A heterogeneous network must be flexible to avoid such inefficiency and inconvenience. It must be able to dynamically assign idle network resources to overloaded traffic. In the best case, a flexible heterogeneous network provides higher performance and is more cost effective.

Other network design issues include reliability, fault tolerance, modularity, expandability, and transparency. Network transparency is a particularly important design issue of a heterogeneous network. A transparent network maintains single network interface for all types of traffic.

1.3. Switching Methods

1.3.1. Circuit Switching

Circuit switching consists of three phases: circuit setup, data transfer, and disconnection. When a call request arrives, a call setup procedure is invoked to reserve a connected series of idle links along the path between the source and destination of the call. A circuit switched network normally operates in blocking mode; if a system fails to find an idle path for a call, the system rejects the call. Once a circuit is established, the circuit is exclusively assigned to the call until it completes. Data is transmitted with minimal processing at

intermediate switching nodes during the data transfer phase. Any signals from both ends (in full duplex cases) including idle signals are transmitted to the opposite ends. After the data transfer phase, a disconnect procedure frees the links comprising the circuit. In early telephone communication systems, physical circuits were manually connected at switching offices. Because circuits were established manually, these telephone systems suffered from long connection setup times. In current telephone systems, a large bandwidth trunk is divided into 4 KHz channels with different base frequencies. A circuit consists of a series of idle 4 KHz channels along a path. Because a circuit usually consists of channels of different base frequencies, each switching office along the circuit must switch the base frequencies of incoming channels to those of outgoing channels. Even if the base frequency adjusting functions are applied at intermediate nodes of a established circuit, circuit switching still provides fast delivery because channels are already reserved (hence, no queueing delay). The constant propagation delay between the two ends of a circuit is the major delay component of circuit switching.

1.3.2. Packet Switching

In a store-and-forward network, transmission lines are dynamically shared by active sessions. A store-and-forward network system operates in delay mode; the system always accepts incoming messages (except during periods of overload), but the messages are not immediately delivered. When a node (either the source or an intermediate node) receives a message, the node checks for transmission errors and selects the outbound link for the message. We emphasize that the entire message must be received before performing any switching functions. The message is queued (or stored) at the selected outbound link if other messages are being served or waiting for the service from the link. Eventually, the message is

transmitted (or forwarded) to the next node following the preceding messages. The message uses the outbound link only when it is actually transmitted. At all intermediate nodes, the same store and forward procedures are repeated until the message arrives at its destination. The error checking and message routing procedures take relatively short time compared to the queueing (waiting at line queues) delay and the transmission (forwarding) delay. However, the queueing delay can vary greatly depending on network conditions. Hence, the total message delay in the store-and-forward network can vary and is larger than that of the circuit switching network.

A store-and-forward network can be operated either by packet switching or by message switching. The unit of transmission in message switching is an entire message. The description of the store-and-forward network in the previous paragraph is based on message switching. Message switching is biased against short messages. If long messages and short messages are assigned the same link access priority, long messages use transmission lines longer than short messages. Short messages experience unduly long delays because they have to wait until preceding long messages are transmitted. Because of this, and the unpredictable demands placed on buffer space at nodes, message switching is seldom used in practice.

Instead of sending a message in its entirety, packet switching partitions the message into several short packets and transmits the packets independently. At the destination node, the packets are reassembled to form the original message. Packets originated from different messages dynamically share transmission lines. Consequently, packets partitioned from the same message may not be transmitted in succession, but can be interrupted by packets from other messages. Packet switching increases the link utilizations because transmission lines are dynamically shared by many users. Also, packet switching alleviates data buffering problems

at intermediate nodes because only small partitions of a whole message are stored. However, packet switching incurs the overhead of message partitioning and reassembling operations. Also, each packet includes overhead bits such as headers and trailers, and this overhead puts extra processing and transmission burdens on switching nodes and transmission lines.

1.3.3. Application Areas of Switching Methods

We mentioned in section 1.1 that each traffic type has its own distinctive characteristics. These distinctive traffic characteristics require different network properties. For example, voice traffic usually does not require error free transmissions, but error protection is mandatory for data traffic. More importantly, it is difficult to efficiently support heterogeneous traffic with a single switching method. Data traffic is characterized by long, low activity ratio sessions. Suppose that we use circuit switching for data communications. Circuit switching would assign a non-sharable circuit to each data session. The circuit is idle most of time because of the low activity ratio of data traffic. Once a circuit is assigned to a data session, the circuit is not available to other users until the session releases the circuit. Hence, circuit switching can support only a limited number of data traffic sessions. On the other hand, suppose we use packet switching for voice traffic. A voice call would be partitioned into smaller packets, including overhead bits. The packetization overhead has a greater impact on voice traffic than data traffic because voice traffic has higher activity ratio. Also, the potential for variable packet delays and the need to reorder packets at their destination make packet switching less attractive for voice traffic.

Performance analysis studies by Miyahara, Hasegawa, and Teshigawara [MiHT75], [MiTH78], Rudin [Rudi78], and Kermani and Kleinrock [KeKl80] have shown that traffic

with different characteristics requires different switching methods. Miyahara, Hasegawa, and Teshigawara [MiHT75] [MiTH78] studied the performance of message, packet, and circuit switching using a queueing model of tandem networks. The performance measures used in the study were message delay and network throughput. They discovered that, for short messages, packet switching provided a faster delivery and higher throughput than the other two techniques. But circuit switching provided better services to bulk data. Kermani and Kleinrock [KeKl80] confirmed the above result; i.e. circuit switching for bulk data and message (packet) switching for short data. In addition to that, they found that packet switching became more attractive in larger networks and at higher workloads. This result is intuitive because the dynamic link sharing advantage of packet switching is more prominent in larger networks (with long paths) at heavy traffic loads. Rudin [Rudi78] performed a cost analysis on switching methods using two types of traffic: continuous messages and intermittent messages. The cost function in the study included the costs of message processing and message transmissions. With continuous messages, packet switching is preferred for short messages and for short distance transmissions, and circuit switching is preferred for long messages and for long distance transmissions. Packet switching is more attractive than circuit switching in intermittent message cases. In addition to short messages, packet switching transmits intermittent messages more effectively than circuit switching if transmission distance is long. Circuit switching wastes the transmission capacity of long circuits because the long circuits are occupied even in during idle periods. However, circuit switching is still more effective for long messages and short transmission distances. Rudin's result indicated that both switching methods had their own application areas.

1.4. The Integrated Network

Because any single switching method cannot effectively support all classes of heterogeneous traffic, one would like to include circuit *and* packet switching in heterogeneous communication systems. A straightforward way to handle heterogeneous traffic is to install two separate networks: a circuit switching network and a packet switching network. The separate network approach is neither flexible nor cost effective. We already mentioned that a communication system carries spare transmission capacity to handle peak demands. If two separate networks are used, each network must carry excess capacity for peak workloads. It is difficult to share the excess transmission capacity because the networks are separately managed. Coviello and Vena [CoVe75] observed that the statistical deviation of the sum of two workloads is smaller than the sum of each workload's deviations. It is possible to handle peak traffic demands with smaller overhead capacity by merging all heterogeneous traffic in one network whose resources are shared dynamically.

In addition to the inflexibility, the separate network approach also lacks transparency. This approach has two network/user interfaces and users must switch the interfaces when they switch applications which use different switching methods.

An integrated (hybrid) switching network supports circuit switching and packet switching in one network. Kummerle [Kumm74] used a time division multiplexing (TDM) method to design a frame structure that simulated circuit switching in a packet switching network. Coviello and Vena [CoVe75] developed a refined structure called the SENET (Slotted Envelope NETwork). A trunk is partitioned into frames each of which occupies the trunk for short time intervals. A frame is further split into slots which are equivalent to channels in terms of circuit switching (circuit slots) and are equivalent to a packet transmission unit in

terms of packet switching (packet slots). The two types of slots are divided by a boundary. Figure 1.1 illustrates the frame structure of the SENET. A circuit switched call occupies one slot per frame continuously until it is disconnected. These slots are reserved as a channel is reserved in circuit switching. Only real data (including idle bits) are transmitted in the circuit slots. Likewise, one packet slot can transmit one packet, including a header and a trailer. Packet switched messages compete and share packet slots.

An integrated network assigns slots to circuit traffic and packet traffic to satisfy a specified grade of service. The grade of service is the blocking probability for circuit traffic and the average message delay for packet traffic. Suppose a frame consists of S circuit slots

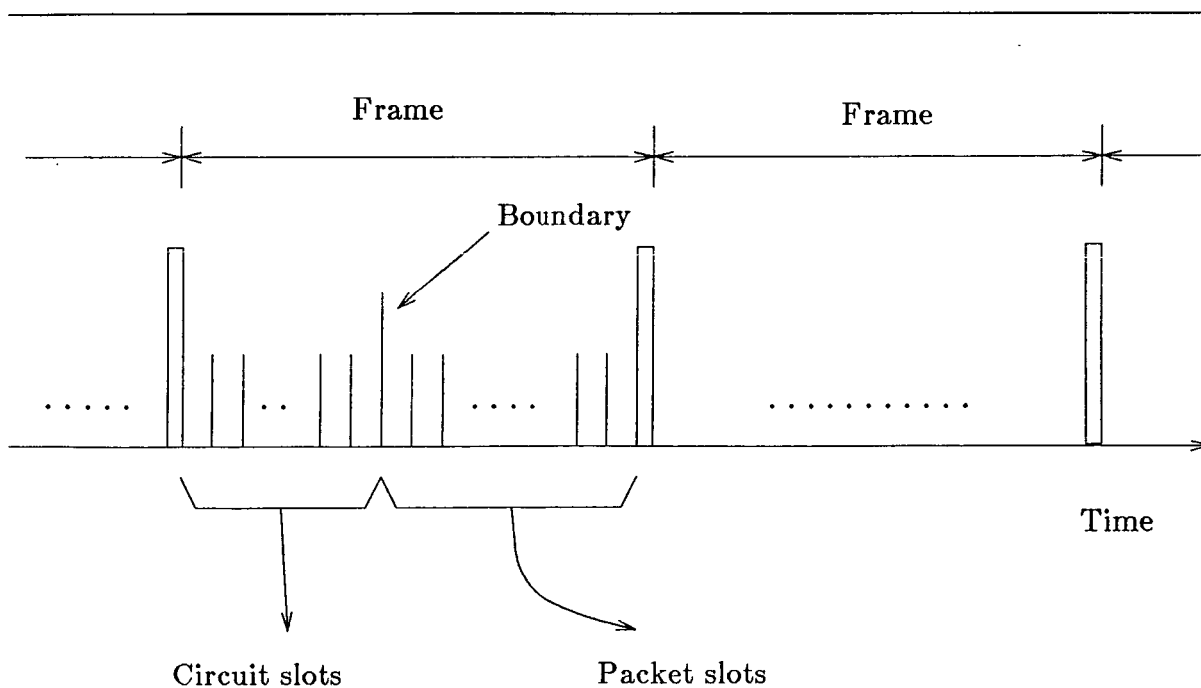


Figure 1.1. Frame structure.

and N packet slots. There are two methods to manage the boundary between the circuit slots and the packet slots. One is a fixed boundary policy where the boundary does not change its position dynamically. Another method, a movable boundary scheme, changes boundary positions dynamically depending on instantaneous traffic mix. In addition to the N packet slots, packet traffic can use unassigned (not reserved) circuit slots. The movable boundary method more nearly achieves the potential advantages of an integrated network. The movable boundary policy uses transmission lines more effectively and provides shorter delay and larger throughput to packet traffic than the fixed boundary policy.

Earlier, we discussed the design issues for networks. Clearly, integrated networks have potential performance advantages. In addition, integrated networks have several benefits over other design approaches.

Flexibility: A movable boundary hybrid network can dynamically assign transmission resources to circuit traffic and packet traffic according to transient and long term traffic mix changes. The flexibility of a hybrid network enhances network performance and provides convenient network management.

Cost Effectiveness: The cost effectiveness of an integrated network is a direct result of flexibility. An integrated network can handle heterogeneous traffic workloads with less transmission capacity than two separate networks. In addition to the transmission resources, heterogeneous traffic can share some switching resources because both switching methods have common switching functions.

Transparency: A hybrid network can provide a single network interface to both switching methods. An unified network interface enhances the ease of system use and encourages the development of applications that use both switching methods.

1.5. Overview of Following Sections

Section 2 contains the detailed descriptions of previous research work on packet, circuit, and hybrid switching. First, two modes of packet switching, datagram and virtual circuit, are discussed and compared. Several packet routing algorithms are examined later. This section also includes the description of several circuit routing methods. A discussion of prior research on integrated networks follows the descriptions of packet and circuit switching. First, we describe the proposed solutions for the integrated switching node design problem. After that, we discuss several analytic and simulation performance studies on hybrid switching. These performance studies show the advantage of the movable boundary scheme over the fixed boundary scheme.

In section 3, we discuss the motivation (goals) and methodology of our performance study on packet routing algorithms. In the first part of the section, we discuss several former performance studies on packet routing algorithms. These performance studies showed the advantage of the adaptive routing methods over static routing methods. After reviewing the previous performance studies, we list the motivations for our performance study. The second part of the section deals with the simulation models used in our study. The general descriptions of the simulation model, including the simulation language and regeneration method, are given in section 3. Finally, the section describes the simulated networks used in our study.

Section 4 deals with the packet routing network simulation. First, we give the detailed descriptions of the simulation workloads for four simulated networks. These descriptions are summarized in four tables. The discussion of the simulation results from the four networks follows the simulation model descriptions. The results indicate that the hybrid routing algo-

rithms provide better performance in all four networks. Especially, the performance of a proposed hybrid weighted routing algorithm is outstanding.

Section 5 contains the proposal for future research. The three main goals of the future research are discussed. The three goals are

- comparison of packet routing methods in a movable boundary integrated switching network with larger packet traffic workloads
- investigation of the performance of circuit routing methods in an integrated switching network and the interaction of the packet and circuit switching parts of an integrated switching network
- implementation of several promising packet and circuit routing algorithms in a real integrated switching network to compare the results from the real system and network simulations. We are planning to implement several routing methods in a NASA's space station integrated switching network [Murr84].

In addition to these research goals, we discuss the research tools necessary for accomplishing these research goals in section 5.

2. PRIOR WORK

Packet switching and circuit switching were briefly discussed in section 1. More detailed descriptions of the previous research work on packet and circuit switching are given here. For packet switching, we describe and compare two modes of packet switching operation: datagram and virtual circuit. After that, we discuss packet and circuit routing algorithms in detail. Several earlier comparisons of packet routing methods will be discussed in section 3. For integrated switching, we first explain proposed approaches to the switching node design issue. We also describe several analytic and simulation performance studies. These studies showed the performance advantage of the movable boundary management scheme over the fixed boundary management scheme.

2.1. Packet Switching

ARPANET [McWa77], which was developed in late 1960's by ARPA (the Advanced Research Projects Agency of the U.S. Department of Defense), was the first operational packet switching network. Many algorithms and protocols of packet switching were developed in the design stage of the ARPANET, and the ARPANET research triggered many research efforts in packet switching network design and algorithm development. Numerous packet switching problems such as congestion control, routing, and flow control problems were addressed and efficient algorithms for these problems were developed.

We begin with a survey of proposed packet switching routing techniques because these algorithms play an important role in integrated switching networks. Before discussing the details of these packet routing methods, we examine classification schemes for these methods.

2.1.1. Virtual Circuits versus Datagrams

Packet switching can be operated in two modes: virtual circuit and datagram. The virtual circuit technique assigns a path to each data session and all packets generated during the session are routed over this path. In datagram packet switching, each packet can potentially follow a different path

2.1.1.1. Virtual Circuits

Virtual circuit routing has connection setup, data transfer, and disconnection phases similar to circuit switching. When a network receives a virtual circuit connection request for a data session, the network assigns a virtual circuit number to the session. The source node of the data session selects an outbound link and registers the virtual circuit number and the outbound link in its routing table. The node sends a control packet for virtual circuit connection over the selected link. This path setup procedure is repeated at intermediate nodes until the destination node of the data session is reached. After the path setup procedure, each node on the selected virtual circuit (including the source and destination nodes) has recorded the outbound links of the virtual circuit in its routing table. Once the virtual circuit is set up, all packets generated from the data session are routed over the assigned path. All packets carry virtual circuit identification numbers. When a node receives a packet, it checks the virtual circuit number of the packet and reads the outbound link of the virtual circuit from its routing table. Unlike circuit switching, the links in a virtual circuit are not exclusively assigned but are shared by other data sessions. Hence, packets travel over virtual circuits in a store-and-forward mode.

Once a virtual circuit is set up for a data session, it would be normally be used until the end of the session. However, if the network status changed dramatically (e.g. link failure) during the data transfer phase of the session, new virtual circuits might be created to avoid broken or severely congested paths. Even if the virtual circuit method could change routing paths (virtual circuits), it cannot change paths quickly because of the time and overhead of old virtual circuit disconnection and new virtual circuit setup. In both cases, a network has to send control packets over the old and new virtual circuits.

2.1.1.2. Datagrams

Virtual circuit routing considers a data session as an unit of connection. Once a path is set up for a data session, all packets generated from the session take the same path. In datagram routing, paths are independently determined for each packet. Even the packets from one message may use different paths. Because a packet is the unit of routing decision, each packet must carry its destination node address. The routing tables used in datagram switching are indexed by destination node addresses. When a packet arrives at the source or at an intermediate node, the node decides an outgoing link by indexing its routing table with the destination node address of the packet. The size of routing tables used in datagram switching is fixed. Generally, these routing tables have fewer entries than those used with virtual circuits because the number of nodes is much smaller than the number of active data sessions in most networks.

It is easy to change routing paths in datagram packet switching networks since paths are determined as packets are transmitted toward the destination. A datagram packet switching network can dynamically update routing tables in case of network status changes

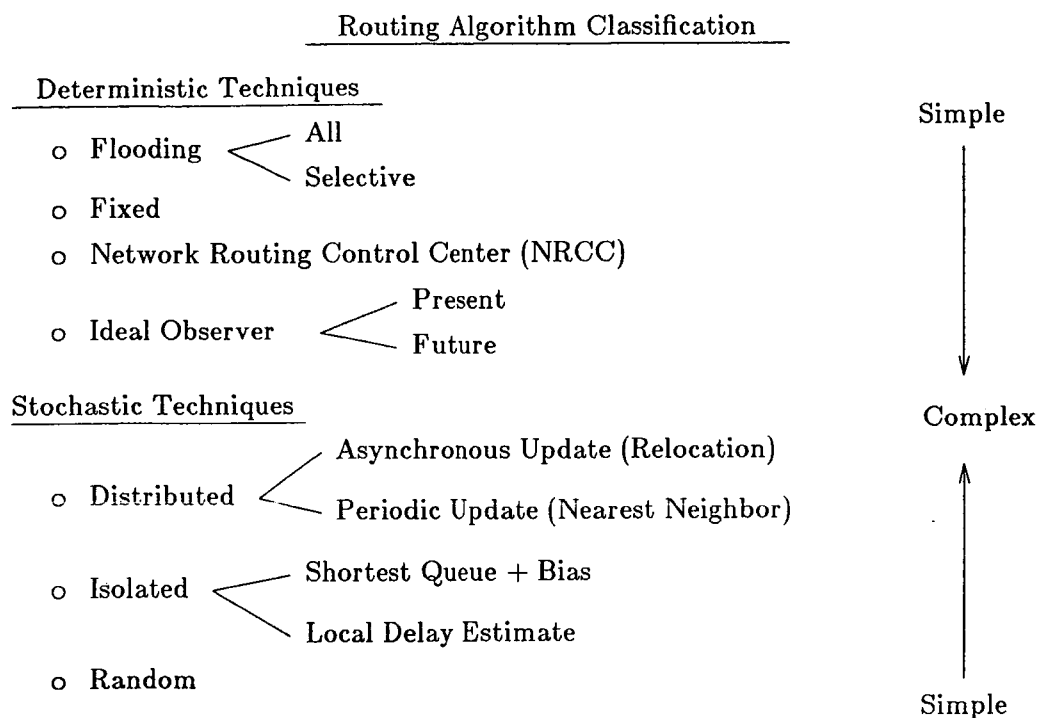


Figure 2.1. Fultz and Kleinrock's Classification.

using adaptive routing algorithms. Adaptive routing algorithms monitor the network status and route packets over most promising paths. Hence, one can improve network performance by using efficient adaptive routing algorithms. We will later discuss adaptive and other nonadaptive routing algorithms in a greater detail.

2.1.2. Classification of Routing Algorithms

Many packet routing algorithms have been developed and used in operational networks. Each algorithm has its own operating principles; however we can classify packet routing algorithms into several groups based on their characteristics. Several classifications of routing procedures were developed using different criteria. We discuss these classifications of routing algorithms before explaining the principles of individual algorithms. By investigating these classifications, we can obtain greater insight into the complexity and principles of packet routing methods [FuKl71].

Fultz and Kleinrock's classification [FuKl71] was the first classification of packet routing algorithms. They grouped packet routing algorithms on the basis of routing decision rules: deterministic techniques and stochastic techniques. Deterministic techniques make routing decisions with fixed rules. These routing algorithms provide loop-free routings (packets do not visit a node twice). Stochastic routing techniques use probabilistic routing rules. For example, random routing chooses outbound links arbitrarily. The other stochastic routing procedures, isolated and distributed routing, are adaptive; they sense and adapt to instantaneous network changes. In distributed routing, nodes report the network status perceived at the nodes to their neighbors. Isolated routing uses only directly estimated or observed network status information in making routing decisions. Figure 2.1, excerpted from [FuKl71], shows this classification.

Later, Rudin [Rudi76] refined the classification of packet routing techniques. He classified routing techniques as centralized or distributed based on the location where routing decisions are made. Except *fixed* routing which starts with externally prepared routing tables, centralized techniques use a designated central node called NRCC (Network Routing

Control Center) that gathers network state information and creates routing tables with the information. Other nodes report network status information to the NRCC and receive routing tables from the NRCC. In distributed techniques, each node makes its own routing decisions either arbitrarily (*random routing* and *flooding*) or based on network state. Figure 2.2, taken from [Rudi76], shows this classification.

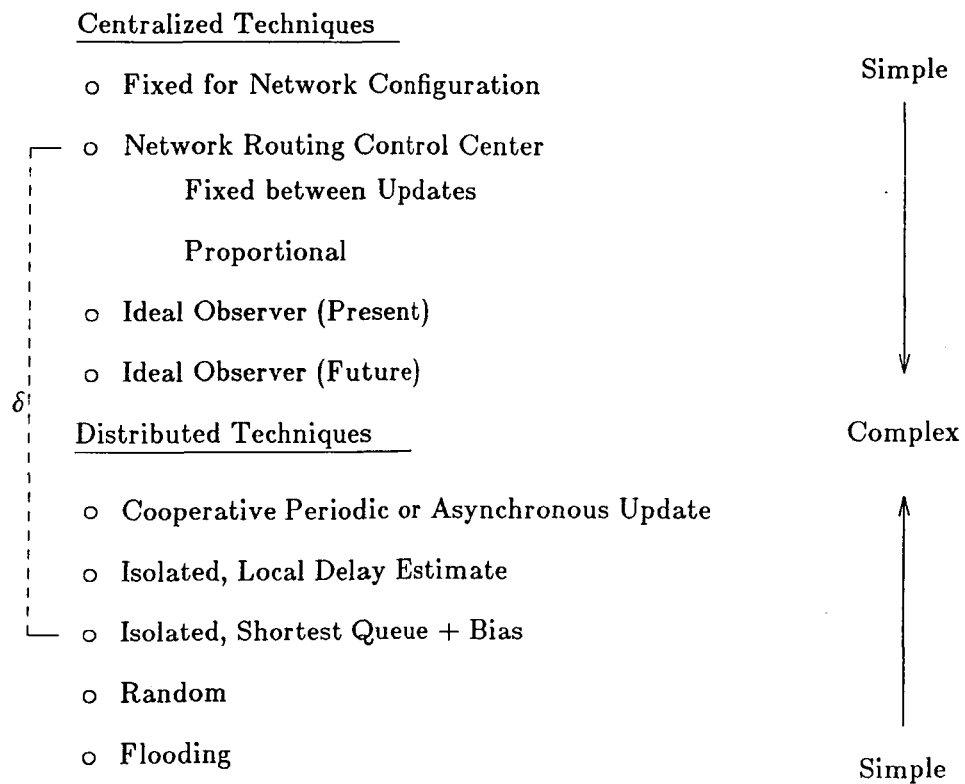


Figure 2.2. Rudin's Classification

Rudin [Rudi76] further classified packet routing algorithms using a three dimensional model. This three dimensional model is shown in Figure 2.3. One dimension divides routing techniques based on the locations where routing decisions are made. The second dimension specifies the type of information used to make routing decisions: local information or global information. Local information is the knowledge of network status which can be observed or estimated directly at a node. For example, information about the states of links incident on the node is the local information of the node. Global information is a collection of network status knowledge transmitted from other nodes. Global information allows a comprehensive

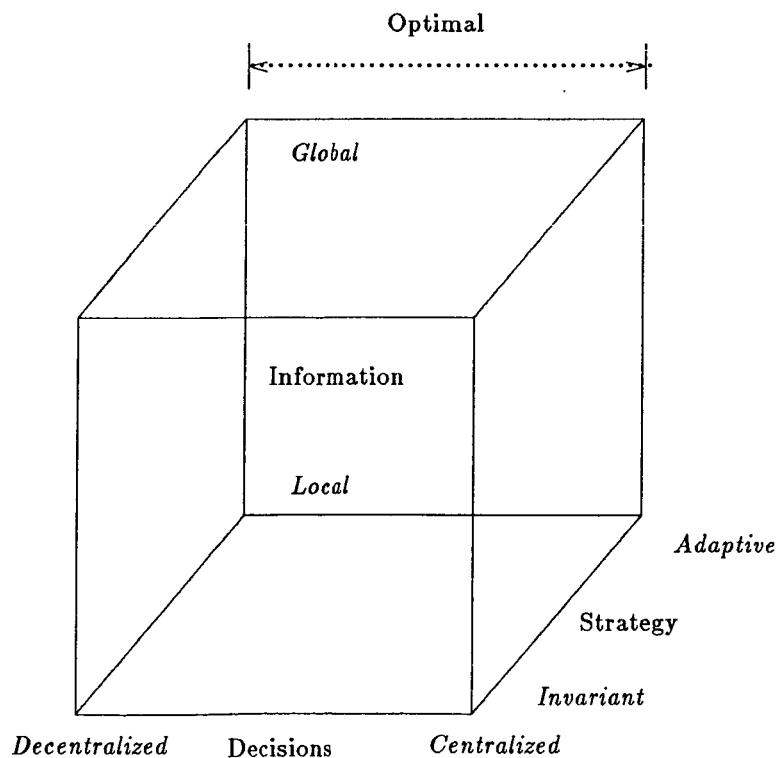


Figure 2.3. Rudin's three dimensional model.

view on network conditions. However, a network must use extra network resources to collect global information. The last dimension is based on the frequency of routing decision updates. At one extreme of this dimension, *fixed routing* uses the same routing tables until new routing tables are externally prepared and loaded into the system. Adaptive routing procedures monitor network status changes and modify routing paths according to the network status.

Centralized routing techniques such as *NRCC routing* use global information to make routing decisions. These algorithms use the same routing tables until new routing tables are created. Hence, these methods do not quickly adapt to instantaneous network changes occurring between routing updates. But these techniques provide stable routing decisions because they make routing decisions based on global information. Distributed routing techniques such as *shortest queue routing* use local information. These routing methods can update routing decisions frequently with readily available local information. However, routing decisions based on local information are usually unstable. *Looping* is the frequent problem of the distributed routing techniques. A hybrid routing procedure combines the two strategies. It has a NRCC that makes routing decisions based on global information. In addition to the NRCC, other nodes participate in routing decisions based on the local information available at the nodes. When the NRCC cannot make clear decisions, it delegates final authority to local nodes. A hybrid technique achieves the stability of the centralized techniques and the adaptability of the distributed techniques.

2.1.3. Description of Packet Routing Techniques

As we have seen, Fultz and Kleinrock [FuKl71] classified packet routing methods based on the properties of decision rules. Conversely, Rudin [Rudi76] developed a new classification

scheme based on the location where routing decisions are made. Rudin also developed a hybrid routing algorithm called *delta routing*. In hybrid routing techniques, both the NRCC and local nodes participate in routing decisions.

In the following sections, we describe several routing methods and classify them as centralized, distributed, or hybrid techniques. The centralized techniques include fixed routing and NRCC routing. For the distributed techniques, we explain the properties of random, flooding, shortest queue, backward learning, and cooperative update routing methods. The hybrid techniques include delta, priority queue, and BS-JBQ routing methods.

2.1.3.1. Centralized Techniques

Fixed Routing (Static routing): In fixed routing, routing tables are externally developed and loaded into the system. One set of routing tables is used until the next routing table set is loaded. Routing tables are generated from the solution of a *multicommodity flow* problem [FrCh71], [FrGK73]. In a communication network, all traffic to one destination node comprises one commodity. If a network has N nodes, there are N commodities in the network. The multicommodity flow problem in a communication network is to minimize weighted queueing delays by optimally assigning flows of commodities to alternative paths. The problem assumes that network configuration and traffic workloads between source-destination pairs (the amounts of commodities) are known and stationary. A solution of the multicommodity flow problem is converted into routing tables for each node. A routing table has entries for each destination (commodity). A routing table entry includes all alternative outbound links to a destination, along with associated probabilities. When a node routes a packet, it generates a random number and selects a path whose probability region includes

the random number. The performance of *fixed routing* is satisfactory in stable networks. But it cannot adjust its routing tables when the network state changes.

NRCC Routing: Unlike *fixed routing* where routing tables are externally prepared, NRCC routing updates routing tables internally at a central node (NRCC). This enhances its adaptability to network status changes. The procedure to update routing tables is as follows. First, nodes report locally perceived network state information to the NRCC when it is necessary to update routing tables. The initiation of a routing update is triggered in one of two ways: periodic updating and asynchronous updating. In periodic updating, the routing update interval is fixed and nodes periodically report network status to the NRCC at every routing update interval, regardless of the network state. In asynchronous updating, nodes report network information only when the nodes observe significant changes (more than a predefined threshold value) in packet delays (or any cost criterion). With the reported global information, the NRCC generates new routing tables and sends the routing tables to local nodes.

A routing table in a local node encodes the single best path from the node to each destination node. Usually, the paths are generated by solving a *shortest path* problem. The *shortest path* problem is to find the minimum cost paths from sources to destinations. The solution of the *shortest path* problem selects only one best path for each source–destination pair. More sophisticated methods choose multiple paths for each source–destination pair. In these cases, the NRCC selects multiple paths and distributes fractions of the traffic to alternate paths. Many solutions such as the flow deviation method [FrGK73], and gradient projection method [ScCh76] have been proposed to solve this multipath routing problem.

Unlike fixed routing, NRCC routing can adapt to network changes automatically. However, NRCC routing does not adapt quickly to instantaneous traffic changes because routing paths are fixed until the next table update. One can enhance the adaptability of NRCC routing either by updating routing tables more frequently (periodic updating) or by using smaller threshold values (asynchronous updating). Still, NRCC routing does not react quickly. Local nodes that sense network changes cannot correct the old routing decisions by themselves. Instead, local nodes must report the changes to the NRCC, where changes are made and sent to the local nodes. In addition to the inadaptability, NRCC routing consumes transmission resources when sending control packets of global information and routing tables. The consumption of transmission capacity is most severe at the links adjacent to the NRCC because all control packets are sent to and from the NRCC. Also, NRCC routing is vulnerable to failures because all routing decision functions are centered at the NRCC. A backup NRCC must be reserved to improve network reliability.

2.1.3.2. Distributed Techniques

Unlike NRCC routing, where all routing decision responsibility is centered at the NRCC, every node makes its own routing decisions in distributed routing techniques. Distributed routing techniques include random, flooding, shortest queue, backward learning, and cooperative update routing. Among these distributed routing algorithms, shortest queue, backward learning, and cooperative update routing methods are adaptive routing methods. With these routing methods, local nodes can respond quickly to instantaneous network changes. The other two methods, random and flooding, route a packet over one randomly selected link or over all outbound links.

Random Routing: When a packet arrives at an intermediate node, the node randomly chooses a link from a set of candidate outbound links. The candidate outbound link set may include either all links incident at the node or only a selected subset of all incident links. In *selective random routing*, candidate links are selected for each destination to eliminate bad paths and to avoid loops. Because random routing is one of the least complex algorithms, it is frequently used as the base routing technique in many network performance studies. The performance of more complex algorithms is then compared to the performance of random routing.

Flooding: When a node receives a packet (assuming the node is not the destination of the packet), it sends copies of the packet over all outbound links or over a selected subset of outbound links. In flooding, a node may receive the same packet more than one time. The node must recognize returning packets and not rebroadcast them. This method is highly redundant and consumes vast amount of transmission resources. However, flooding provides very reliable transmissions because multiple copies are sent over multiple paths. There are several application areas where flooding routing is appropriate: a military communication system in an extremely hostile environment or concurrent updates of distributed databases.

Shortest Queue Routing When a node receives a packet, the node compares the queue lengths of outbound links and sends the packet over the link with the shortest queue length. Shortest queue routing may restrict the candidate links depending on destinations to avoid apparently *bad* paths. Because a node does not know the global network status, it may send a packet over a link which eventually leads to a bad path. Also, this method has a *looping* problem. However, shortest queue routing has neither the transmission overhead nor processing overhead occurred in routing methods that use global information.

Backward Learning: A node updates its routing table based on the delay experienced by packets arriving at the node. Suppose node i has L incoming lines l_1, \dots, l_L . Packets originated at node j can arrive at node i over any of the incoming lines. The packets are timestamped at node j and node i calculates packet delay by subtracting the departure time from the current time. Node i remembers only the minimum delay of incoming lines. Let the minimum delay on line l_k be D_{jk} . If d_{jk} , the delay of a packet which arrives at node i over link l_k , is smaller than D_{jk} , then d_{jk} replaces D_{jk} . Node i sends packets destined to node j over the link with the smallest minimum delay. Because the minimum delay of a link is updated only when newly arrived packets over the link have smaller delay is smaller delay than the current minimum delay, it retains a small value once set to the value. Hence, node i may route packets over a bad path which once was the best path. To avoid the problem, node i periodically sets minimum delay of outbound links to a large value.

Cooperative Update Routing: In cooperative update routing, each node keeps routing tables that record the selected paths and their associated transmission costs for each destination node. A delay table records only the transmission costs. Periodically (periodic method) or asynchronously (asynchronous method), nodes exchange delay tables with their adjacent nodes. Suppose node i has three neighbors k, l, m . Node i easily measures or estimates the transmission costs to these neighboring nodes. Let these costs be q_{ik}, q_{il}, q_{im} . Node i creates a new delay table with these estimated transmission costs to the neighbors and the delay tables received from its neighbors. Let $C_k(m)$, $C_l(m)$, and $C_m(j)$, which are recorded in the delay tables reported from node k, l, m to node i , be the transmission costs from node k, l, m to node j respectively. Node i updates its transmission cost to node j ($C_i(j)$) by the following rule

$$C_i(j) = \min (q_{ik} + C_k(j), q_{il} + C_l(j), q_{im} + C_m(j)).$$

Also, the next link entries of node i 's routing table are updated accordingly. In periodic updating, delay tables are exchanged and new routing tables are created periodically. In the asynchronous mode, a node sends its delay table to neighborhood nodes only when the node senses significant changes in transmission costs to other nodes (more than a threshold value).

2.1.3.3. Hybrid Techniques

Route selection responsibility is centered at the NRCC in the centralized techniques, while it is equally distributed to all nodes in the distributed techniques. Hybrid routing methods combines the two routing strategies. In hybrid routing methods, the NRCC makes stable routing decisions with global information and local nodes adapt to instantaneous traffic changes with local information. Rudin [Rudi76] first proposed a hybrid routing method called *delta routing*. Other hybrid routing methods were developed later by Boorstyn and Livne [BoLi81], and Yum and Schwartz [YuSc81].

Delta Routing: Delta routing exploits both global information and local information. A designated central node (NRCC) collects global information and selects paths based on global information. In NRCC routing, the NRCC is the only node responsible for routing decisions. In delta routing, the NRCC decides routing paths only if it can find clearly better paths based on global information. Otherwise, it selects multiple *good* candidate paths and delegates the final routing decisions to local nodes. Local nodes then select the shortest queue links from the candidate paths.

An important parameter of delta routing is δ . Suppose there are m_{ij} candidate paths, whose first links differ from each other, from node i to node j . Let the set of candidate paths

selected by the NRCC for pair (i, j) be I_{ij} and the set of discarded paths be O_{ij} . Suppose that path S is the shortest path of pair (i, j) and C_S is the cost of path S . If path A belongs to the selected path set I_{ij} , then $|C_A - C_S| \leq \delta$ where C_A is the cost of path A . This relation means that the NRCC selects the best path (path S) and other relatively good paths based on global information. On the other hand, the NRCC discards a path B (B is in O_{ij}) if $|C_B - C_S| > \delta$. Those selected paths are sent to local nodes as routing tables. Suppose only one path is selected for a certain destination, then all packets destined to the node will be routed along the path. On the other hand, if more than one path are selected, then a local node selects a path from the candidate paths based on the *instantaneous local information*. The parameter δ coordinates routing responsibility of the NRCC and local nodes. Also, it controls the relative importance of global and local information.

Priority Queue Routing: Boorstyn and Livne [BoLi81] proposed another hybrid routing method based on a queueing model of local nodes. Suppose a node has two outgoing links. Some traffic is routed to either one of the links exclusively (non-bifurcated traffic). Other traffic can be routed over both links (bifurcated traffic). Delta routing and priority queue routing handle the bifurcated traffic differently. In delta routing, the node selects a link using shortest queue routing as soon as a bifurcated packet arrives at the node. Shortest queue routing balances the queue lengths of the two outgoing links. However, if non-bifurcated packets arrive at the node after the queue balance, the queue balance would be disturbed and anomalous use of links (one link has an excessive queue while the other link is idle) would happen at the node. To reduce the probability of such anomalous link use, priority queue routing assigns a lower priority to the bifurcated traffic. The bifurcated traffic is being transmitted only when either one of two links is idle.

Suppose that the packet arrival and departure processes of a link are Markovian. If the bifurcated packets are randomly routed, then each link operates as an independent M/M/1 queueing system. In priority queue routing, the node behaves as an M/M/2 queueing system because all bifurcated packets are stored at one lower priority queue. Boorstyn and Livne claimed that if bifurcated traffic composed 15–20 percent of the total traffic at a node, the node would behave almost like one M/M/2 queue instead of two independent M/M/1 queues.

JBQ-BS Rule: Yum and Schwartz [YuSc81] proposed a hybrid routing method which used *best stochastic (BS)* at the NRCC and *join biased queue (JBQ)* at local nodes. The BS rule distributes fractions of traffic to multiple paths. Normally, the BS rule operates stochastically; local nodes assign packets to links stochastically using the flow distribution rules determined at the NRCC. Yum and Schwartz found that network performance was enhanced if deterministic rules such as shortest queue routing were used at local nodes. However, if shortest queue routing is used, then the resulting traffic flow pattern does not follow the flow distribution rules specified by the BS rule. A bias term (join biased queue) is included to shortest queue routing to route packets according to the flow distributions given by the BS rule.

2.2. Circuit Switching

We discussed the prior studies on packet switching in the previous section. In this section, the other part of hybrid switching, circuit switching, is discussed. First, we discuss control data signaling methods for circuit switching. The circuit switching routing algorithms are discussed later.

Circuit switching uses control data when it sets up and disconnects circuits and transmits network management messages. Control data can be transmitted either by using a small portion of voice channels (channel-associated signaling) or by using a separate signaling channel (common-channel signaling). Common channel signaling (or common channel interoffice signaling) provides faster signaling because larger bandwidth is assigned for control data transmissions. Let us take PCM as an example [Mart76]. PCM converts an analog voice signal to a digital signal. It samples the amplitude of voice signal and quantizes the amplitude using seven bits (128 levels). The human voice spectrum requires 8,000 samplings per second. Each sample consists of seven bits of encoded voice and one extra bit. These extra bits are used to transmit control messages in channel-associated signaling.

Usually, circuit switching is operated in a blocking mode; a call is served only if there are idle connections. When a node receives a call setup message, it invokes a circuit routing procedure to establish an idle connection for the call. Routing table entries are indexed by destinations. A routing table entry for a destination sequentially lists the links which lead to the destination in the order of preference. The node checks the destination of the call and sequentially searches the routing table entry of the destination for an idle link. When the node finds an idle link (actually, an idle channel in a trunk), it reserves the channel and forwards the call setup message to the next node. If there are not any idle links, the node sends back the call setup message to the previous node. The previous node resumes the call setup procedure starting from the node. When the call setup message reaches its destination, a connection message is sent to the source through a reserved circuit. The call is rejected when all alternative paths are busy.

Conventional telephone networks use fixed routing where the ordering of alternative paths is fixed. Girard and Hurtubise [GiHu83] studied adaptive routing methods for the call setup procedure. They compared delta routing, backward learning routing, and fixed routing. In delta routing, the NRCC collects the residual capacity (idle transmission capacity) of each link. Based on the residual capacity, the NRCC selects a set of good paths. Backward learning routing uses the blocking probability of each incoming link as the cost function. Links are listed in the order of increasing blocking probability. They found that delta routing was marginally better than fixed routing. Lippman [Lipp83] also studied adaptive and static routing methods in a mixed media network (a network with terrestrial lines and satellite lines). His study result revealed that adaptive routing algorithms is better than static routing methods when network state fluctuates.

2.3. Hybrid Networks

The previous two sections explained the properties and the routing algorithms of packet and circuit switching, the two components of integrated switching. This section summarizes the prior studies on integrated switching. We classify this work in two ways: 1) integrated network implementation schemes and 2) performance evaluations. Other research efforts include optimal integrated network designs [HsGO78], [GiHO81] and integrated network protocol developments [ZaPK76], [PKRJ76]. First, we explain the proposed integrated network implementation schemes. Later, we discuss the analytic and simulation performance studies on integrated switching.

2.3.1. Network Implementations

The frame structure of integrated switching was first introduced by Zafiropulo [Zaf74]. Coviello and Vena [CoVe75] proposed a refined frame structure called SENet (Slotted Envelope Network). In Zafiropulo's scheme, as shown in Figure 2.4, circuit slots and packet slots can appear at any positions and they are intermingled with each other in one frame. Also, one channel may be partitioned into several segregated slots in a frame. But, once circuit slots are assigned to a circuit, their positions in succeeding frames are fixed until the circuit is released. Zafiropulo introduced a list structure to manage circuit slots. A list element denotes the attributes of a circuit slot such as starting position of the slot, size of the slot, and its call identifier. List elements which specify the same circuit are serially connected. Because circuit slots can appear at any positions, a mechanism must be provided to detect circuit slots. A multiplexor (MUX) uses an associative memory (AM) whose rows specify the starting bit positions, lengths, and circuit identifiers of circuit slots. Figure 2.5 shows the structure of the AM. A circular bit counter increases at a base trunk speed and the counter is compared with the starting positions of AM rows. If there is a matching slot, the buffer address (actually the circuit identifier is read and decoded to the buffer address) of the matched slot is read from the AM, and the MUX transmits the contents of the buffer as many bits as the size of the slot. In case of no match, packets (or idle bits if the packet queue is empty) are sent until the next call slot. This scheme uses slot allocation/deallocation procedures that are similar to main memory allocation/deallocation procedures. The slot allocation/deallocation procedures use the list structure to denote idle and busy slots.

Coviello and Vena [CoVe75] proposed a SENet where a circuit occupies only one circuit slot per frame, and all circuit slots are collected together. The frame structure was illus-

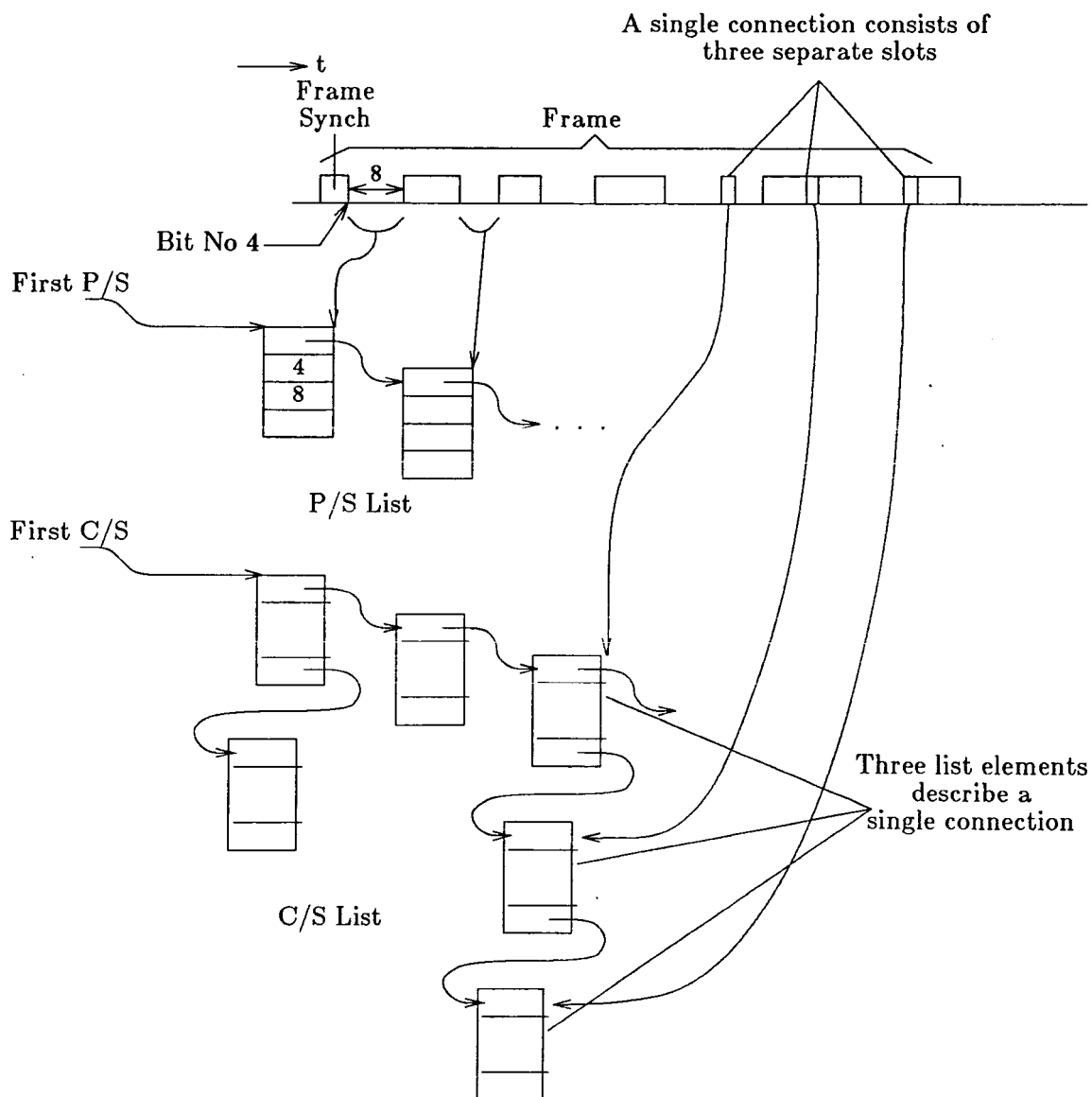


Figure 2.4. The frame and list structures of Zafiropulo's design.

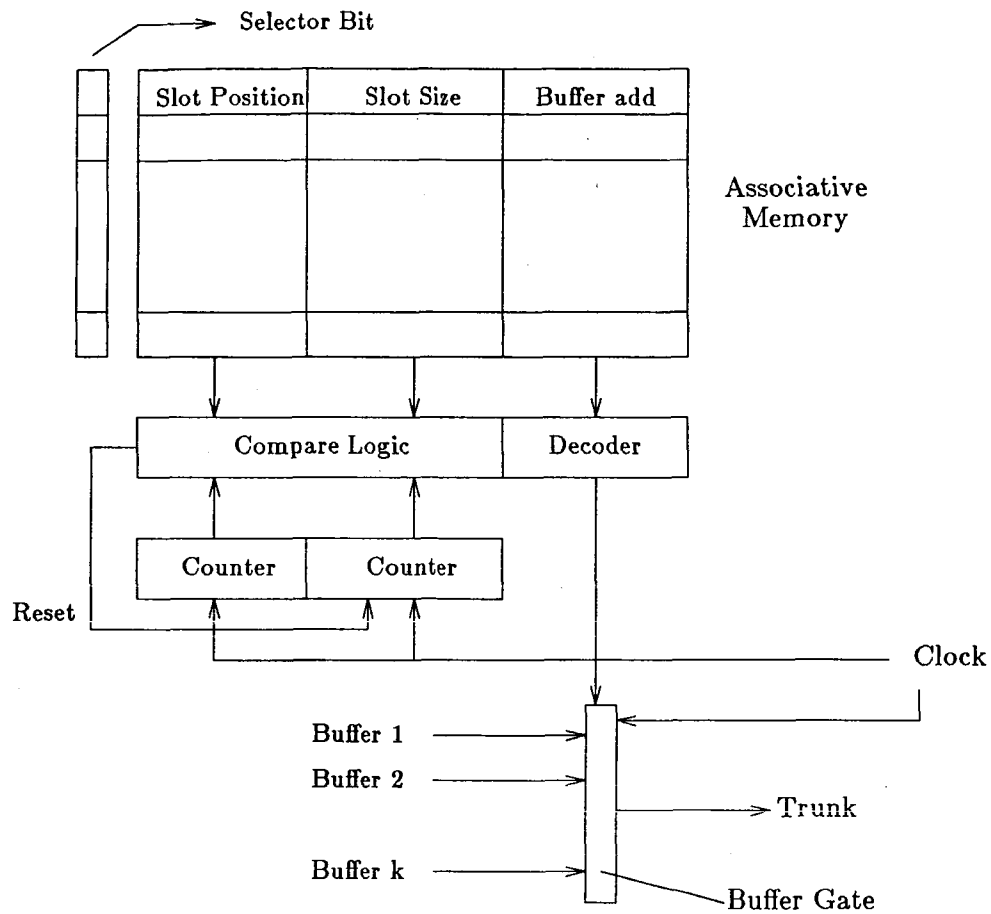


Figure 2.5. Multiplexor structure of Zafiropulo's design.

trated in Figure 1.1. When a request for a circuit connection arrives while a frame is being transmitted, the first idle circuit slot following busy circuit slots is assigned to the call at the next frame. If a call occupying the i -th slot is disconnected, then the slots following the i -th slot advance their positions by one. Consequently, the oldest active call eventually moves to the first circuit slot.

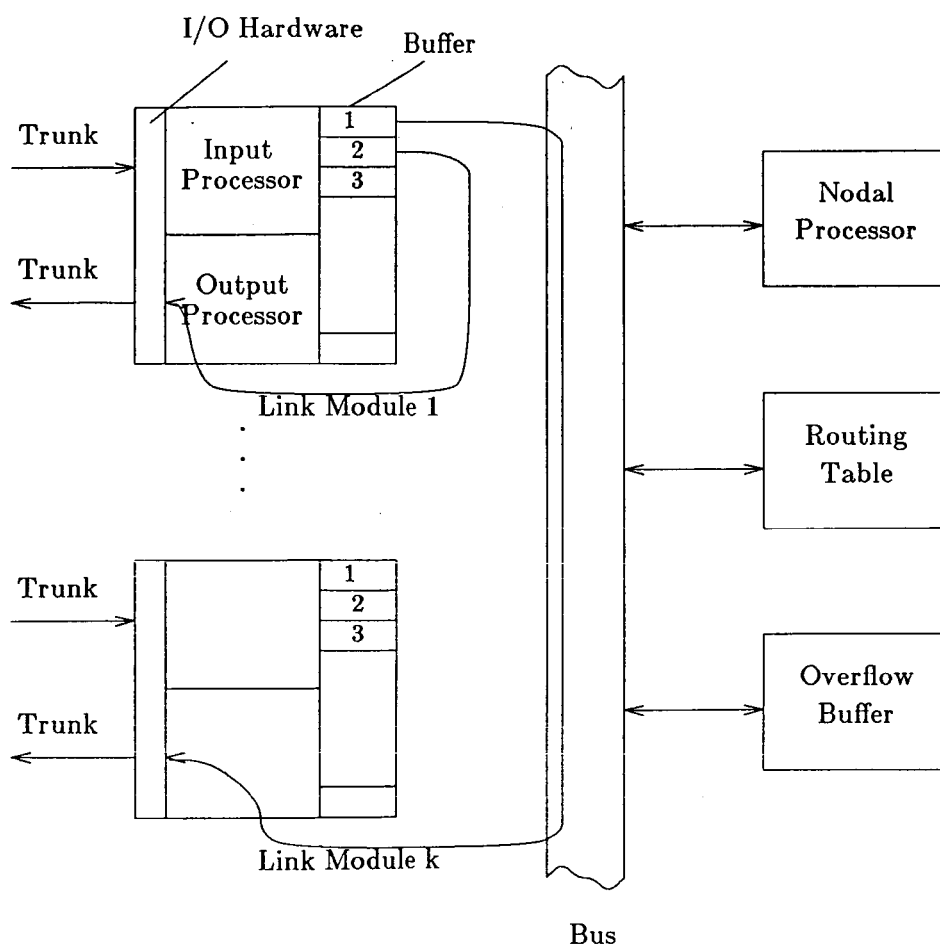


Figure 2.6. Switching node of Ross *et al.*'s method.

While receiving a frame, a node uses a demultiplexor (DEMUX) to separate packet and circuit traffic from the frame. The DEMUX separates slots from the frame and sends circuit switching traffic to buffers at appropriate outgoing lines. The DEMUX must remember the outlinks of established circuits. The DEMUX sends packets to a packet processor. The packet processor checks transmission errors and decides outbound links. Packets are eventu-

ally sent to appropriate trunk queues. A multiplexor (MUX) generates frames from circuit and packet traffic using an associative memory (AM). The structure and the operation of the AM is similar to those used in Zafiropulo's method [Zaf74]. The allocation/deallocation procedures of the SENET is very simple because circuit switching slots are collected together and a circuit consists of only one slot per frame.

The heterogeneous traffic supported in hybrid switching requires dynamic packet and circuit switching functions. We described earlier the list structure and the associative memory used at switching nodes. Ross *et al.* [RoTW77] presented a more comprehensive switching node design scheme for integrated networks. In their proposed design, a switching node consists of a nodal processor, link processors for each incident link, and a redundant bus. The structure of a switching node is shown in Figure 2.6. A nodal processor controls node operations and collects network data. A redundant bus system connects the nodal processor and distributed link processors. A link processor has its own local buffer space and is attached to each full duplex link. A link processor can read from all buffers spread in a node but can write only to its own buffer space. The step by step description of the packet switching function at a node is as follows. First, an incoming packet to the node is stored in the local buffer of its receiving link processor. The receiving link processor checks transmission errors and decides on an outbound link. The link processor at the outbound link is then informed of the buffer address of the packet. The outbound link processor retrieves the packet using the bus and transmits the packet over the outbound link.

Jenny and Kummerle [JeKu76] proposed a similar approach. The structure of their proposed switching node is shown in Figure 2.7. They separated the switching functions, which are applied only to packet switching traffic. When a circuit is established, the outbound links

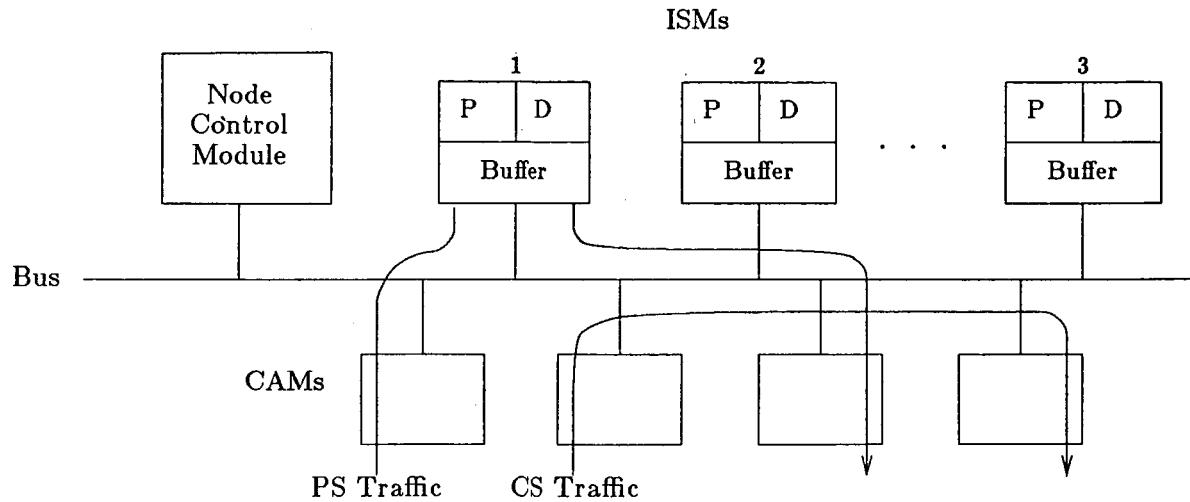


Figure 2.7. Switching node of Jenny and Kummerle's design.

of the circuit are determined at the nodes along the circuit. Circuit traffic just passes through nodes to its predetermined outbound links without receiving switching functions such as error checking, or packet header/trailer regenerating. Communication access modules (CAM), which attach at each line, handle network interface functions. Circuit traffic receives only CAM's services at switching nodes. On the other hand, packet switching traffic requires extra packet handling services. This extra packet handling includes buffer allocation, error checking, acknowledging, and packet routing functions. Intermediate storage modules (ISM) perform the packet-handling process. An ISM has two subprocesses: packet processing and packet dispatching subprocesses. The packet processing (P) subprocess performs buffer allocation/deallocation, error checking, and acknowledgement handling. An ISM performs these functions without interfering other ISM's. The packet dispatching (D) subprocess

requests packet slots on outbound links and passes the buffer addresses of packets to the CAMs at outbound links. The D subprocess in one ISM interfere with other D subprocesses because they compete for shared outbound links.

2.3.2. Performance Analysis

The introduction of the implementation schemes of the integrated switching networks explained in 2.3.1 triggered many performance studies of integrated switching networks [Kumm74], [FiHa76], [OGHF77], [WeMF79], [WeMF80], [MaSc81], [KoPi84], [WiLe84], and [Lian85]. These performance studies dealt with the TDM frame structure in one link, integrated networks. The main research goal of these performance studies was to measure the performance advantage of a movable boundary policy over a fixed boundary policy. Other common issues were packet flow control mechanisms and dynamic voice digitization rate (VDR) control mechanisms to avoid an excessive queue buildup during the periods of peak voice traffic.

Before explaining each performance study, we first describe the parameters and the frame structure of an integrated network commonly used in those performance studies. Maglaris and Schwartz's performance study [MaSc81] dealt with a variable size frame structure. All other studies assumed the fixed size frame whose length is fixed at b seconds on the base trunk of M bits/sec bandwidth. The capacity of one frame is $M \cdot b$ bits. For example, the capacity of a 10 msec frame on the $T1$ carrier (the bandwidth of the $T1$ carrier is 1.544 Mbits/sec) is 15,440 bits. Each frame consists of S circuit slots and N packet slots. Generally, the packet slot is larger than the circuit slot and one packet slot can contains I ($I \geq 1$) circuit slots. The packet arrival process is a Poisson with rate λ_d . Packets have a fixed

length which is same as the length of a packet slot. The call arrival process also follows a Poisson process with rate λ_v and call holding time has a negative exponential distribution with mean $\frac{1}{\mu_v}$.

The performance measures of interest are blocking probabilities (BP) for circuit switched traffic and expected waiting time (EW) for packet switched traffic. Fischer and Harris [FiHa76] have shown that the frame structure of integrated networks has a negligible effect on voice arrival and service processes if the frame length b is small (hence small $\lambda \cdot b$). By ignoring the effect of the frame structure, they obtained the blocking probability from the Erlang B equation. The Erlang B equation is

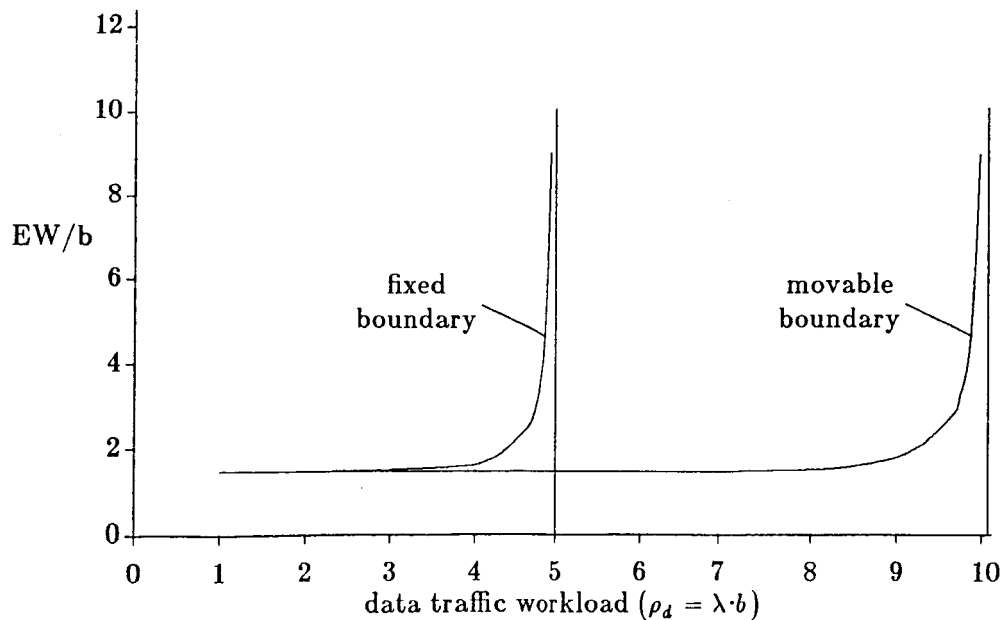


Figure 2.8. Movable versus fixed boundary.

$$BP = \frac{\frac{\rho^S}{S!}}{\sum_{i=0}^S \frac{\rho^i}{i!}} \quad \text{where} \quad \rho = \frac{\lambda_v}{\mu_v}. \quad (3.1)$$

But, it is not straightforward to calculate the expected waiting time of packet traffic if the movable boundary policy is used because packets can be transmitted through idle circuit slots in addition to packet slots.

Fischer and Harris [FiHa76] used an analytical method to calculate the expected waiting time. First, they derived the state probability of voice traffic (state is determined by number of active calls). They then derived the conditional packet traffic state (packet traffic state is defined as the number of packets waiting for transmissions) probability of the current frame. This depends on the circuit and packet traffic states of the previous frame. They unconditioned the packet state probability with respect to previous circuit and packet traffic states. The generating function of the unconditioned packet state probability was then derived. The average length of a packet queue is calculated from the generating function of the packet state probability. Figure 2.8., which is adopted from [FiHa76], illustrates the performance gains of the movable boundary policy. The frame used in this example consists of 10 circuit slots and 5 packet slots. The blocking probability of the network is 0.0223 when the network is loaded with a voice traffic of 5.0 erlangs. They fixed the voice traffic and varied the data traffic workload. The network in their study is saturated when packet traffic intensity ($\rho_d = \lambda_d b$) increases to 5.0 in the fixed boundary policy. The network saturation occurs at $\rho = 10.09$ with a movable boundary.

Occhiogrosso *et al.* [OGHF77] derived the average number of circuit slots available for packet transmission. Let $\rho = \frac{\lambda_v}{\mu_v}$ be voice traffic workload of a system. With ρ and S , we can derive the blocking probability (BP) of voice traffic. If the system serves all voice calls,

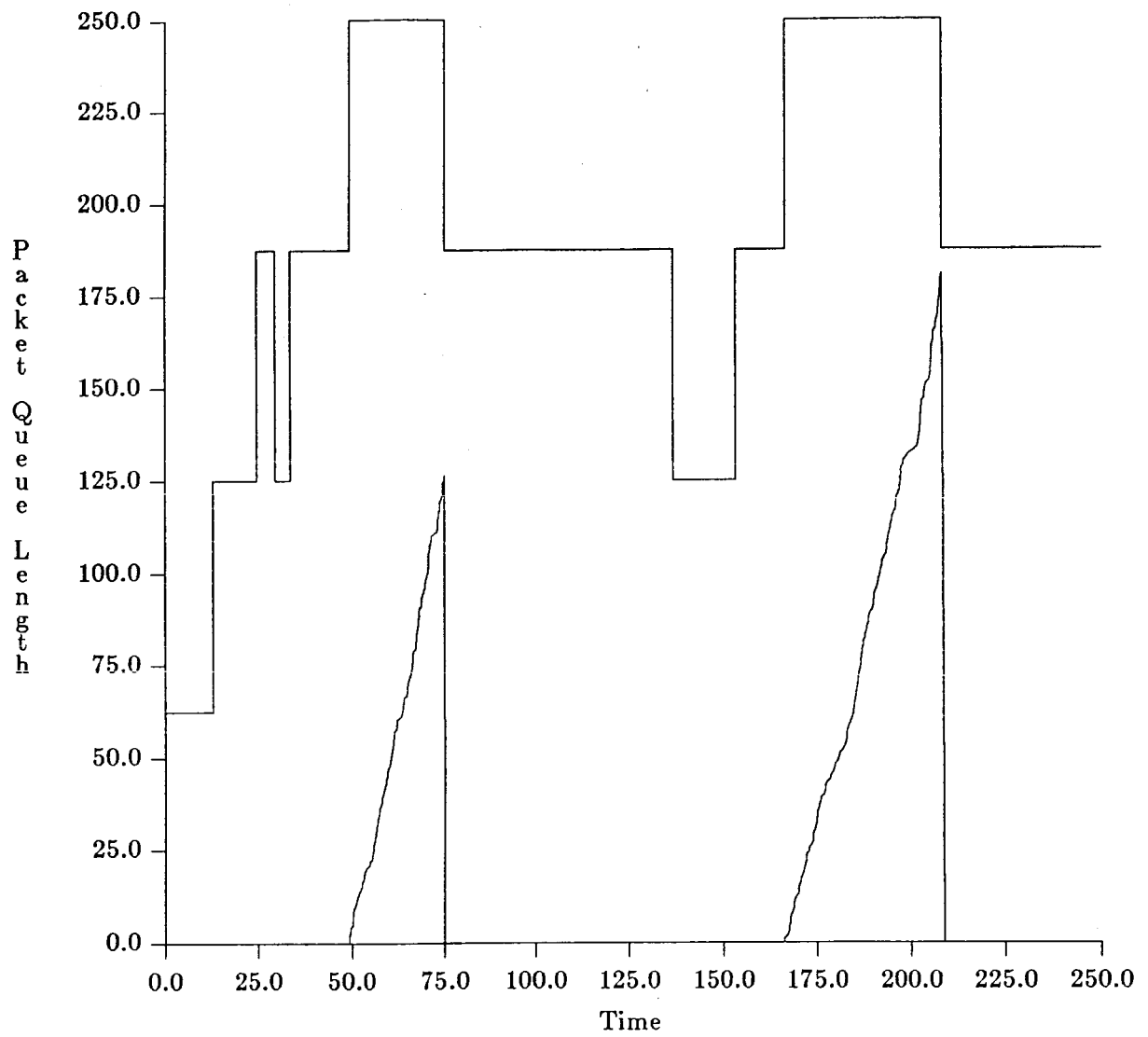


Figure 2.9. Time Varying Packet Queue Length for an Isolated Link
(One Hybrid Packet/Circuit Switched Channel)

there are ρ active calls on average. However, because voice calls are blocked with the probability BP , the number of active call is $\rho(1 - BP)$. Hence, on average $S - \rho(1 - BP)$ circuit slots are available for packet switching traffic. They solved for the expected waiting time using the fact that the average number of circuit and packet slots available for packet traffic is $N + \left\lceil \frac{S - \rho(1 - BP)}{I} \right\rceil$. They derived packet state probabilities assuming that the average packet slots were available at all frames.

Weinstein *et al.* [WeMF79], [WeMF80] found that excessive packet queues would develop if a packet arrival rate is larger than assigned packet transmission capacity (i.e. $\lambda_d \geq N$). Their simulation result (The similar result from Reed's simulation [Reed84] is shown in Figure 2.9) revealed that peak voice traffic periods and excessive packet queue buildup periods were highly correlated. As shown in Figure 2.9, voice traffic states change much more slowly than packet traffic states. Once voice traffic is overloaded, the overloading condition lasts a very long time compared to packet interarrival times. Packet traffic can no longer use circuit slots during the period of high circuit traffic, and an excessive packet queue develops if the packet traffic workloads are larger than the originally assigned packet transmission capacity.

Williams and Leon-Garcia [WiLe84] used a two dimensional Markov chain to denote the states of an integrated switching network; one dimension for packet traffic and another for circuit traffic. Earlier, Weinstein *et al.* [WeMF79], [WeMF80] demonstrated that the two dimensional Markov chain correctly described the behavior of excessive packet queues in circuit traffic overload conditions. All those studies that used the two dimensional Markov chain assumed that the arrival and service processes of the packet traffic are Poisson processes. From the two dimensional Markov chain, transition probabilities and a set of

two-dimensional difference equations for circuit and packet traffic states were developed. Williams and Leon-Garcia developed efficient methods to solve the vector difference equations.

3. MOTIVATION AND METHODOLOGY OF PACKET ROUTING PERFORMANCE STUDY

3.1. Introduction

Several packet routing algorithms and routing classification schemes were introduced in the previous section. We also classified the routing algorithms as centralized, distributed, or hybrid techniques. We selected five routing algorithms from these three routing algorithm groups for our packet routing performance study: NRCC from the centralized techniques, random and shortest queue routing from the distributed algorithms, and delta and priority queue routing from the hybrid techniques. In addition to the five existing methods, a new hybrid routing method called *hybrid weighted* routing was developed and included in our simulation study. Our goal is to extensively analyze the performance of these routing methods in packet switching networks.

In the first part of this section, we discuss the previous performance studies on packet routing algorithms. These performance studies showed the advantage of the adaptive routing methods over the static routing methods. After reviewing the results of these studies, we introduce a new hybrid routing method, *hybrid weighted routing*. The development, motivation, and operating principles of hybrid weighted routing are explained. The motivation of our extensive packet routing performance study is discussed later. Finally, we discuss the simulation model used in this packet routing performance study. The four simulated networks and a routing update implementation scheme are explained.

3.2. Previous Performance Studies

Adaptive packet routing algorithms monitor network conditions and route packets over the most promising paths. In a network where network conditions fluctuate dynamically, adaptive routing techniques can provide users better services compared to static routing methods. Results of previous performance studies on packet switching routing algorithms showed the advantage of adaptive routing methods [FuKl71], [Rudi76], and [ChBN81]. These performance studies showed that adaptive routing methods provided shorter message delays and greater network throughput. The network performance gains from adaptive routing algorithms are converted directly to the cost effectiveness of network systems. Network systems can handle the same workloads with smaller transmission capacity by using adaptive routing procedures. In addition to the cost effectiveness of network systems, adaptive routing methods enhance the fault tolerance of systems because adaptive routing avoids faulty lines and nodes. Also, they improve the reliability and the availability of network systems.

Fultz and Kleinrock [FuKl71] compared the performance of routing techniques using an ARPANET-like network. They used the average message delay as the performance measure of their computer simulation study. They concluded that adaptive routing procedures such as shortest queue and distributed routing provided good network performance under traffic variations. Chou *et al.* [ChBN81] also compared the performance of adaptive and fixed routing techniques by computer simulations. The simulation network used in their study consists of ten nodes and twelve full duplex links which connect the nodes irregularly. The adaptive routing method used in this study was basically shortest queue routing. However, they changed the cost function of shortest queue routing. In addition to various first order cost functions with different coefficients, they included several second order cost functions. They

focused on the properties of traffic and classified network traffic as 1) balanced stationary traffic; 2) balanced traffic with sudden traffic surges; 3) unbalanced traffic with low or moderate workloads; and 4) unbalanced traffic with heavy workload. For the balanced traffic (traffic type 1 and 2), fixed routing is as good as or marginally better than the adaptive routing. Well chosen fixed paths are good in these cases because they are optimized for the particular stationary traffic patterns used in the simulation. However, fixed routing cannot adapt to network traffic fluctuations presented in type 3 and 4 traffic. The adaptive routing selects paths adaptively depending on varying traffic patterns and enhances network performance.

Rudin [Rudi 76] proposed delta routing and compared delta routing with other routing techniques such as random, NRCC, and proportional routing. Proportional routing is a simpler form of bifurcated routing where traffic is distributed over several alternative paths. First, he studied the performances of routing procedures with four small networks. His simulation result showed that delta routing handled traffic fluctuations better than other adaptive or static routing techniques. He extended his study to a ten node irregular network. Delta routing is marginally better than other routing procedures in this medium size network.

Rudin's performance study [Rudi76] compared the performance of centralized (NRCC, proportional routing), distributed (random routing), and hybrid (delta routing) routing techniques. However, adaptive distributed routing algorithms such as shortest queue routing was not included in his study. Furthermore, other hybrid routing methods were developed after his performance study. The performance comparison of these hybrid routing methods has not yet been done.

Boorstyn and Livne [BoLi81] developed a hybrid routing method called *priority queue* routing. And they performed queueing model analysis and computer simulations of switching nodes to show the performance advantage of priority queue routing. They extended the result obtained from the switching node model to complete network systems. First, they found all shortest paths (in terms of number of hops) for each source–destination pair. The fractions of traffic routed over these alternative paths are determined by using several approximation techniques. The queueing delay at each node is calculated with the estimated workloads. They found that priority queue routing outperformed fixed routing. The performance advantage of priority queue routing is more prominent when more traffic has alternative paths.

3.3. Hybrid Weighted Routing

The two hybrid routing methods explained in the previous section (delta routing, and priority queue routing) use global path cost information to select *good* paths at the NRCC. Except this global path selection procedure at the NRCC, global information is not considered in the routing decision process at local nodes. Local nodes route packets based only on directly available network information. These methods must carefully separate *good* paths from *bad* paths at the NRCC. However, it is difficult to identify strictly *good* paths at the NRCC because traffic loads can change very dynamically; old global information becomes obsolete and does not correctly reflect network conditions. If the NRCC chooses paths very generously (in delta routing, select many paths using a large δ), *bad* paths are included in candidate paths. These bad paths can be selected at local nodes because the links of *bad* paths incident at the local nodes may have the shortest queue. On the other hand, if we eliminate too many paths, then we may lose the advantage of a bifurcated routing scheme. In the

extreme case of $\delta = 0$, the two hybrid routing methods degenerate to NRCC routing. The problem here is selecting *good* and *potentially good* paths while discarding *bad* paths.

We developed a new hybrid routing method called *hybrid weighted* routing to solve the problem. To avoid the problem of choosing *bad* paths at local nodes, hybrid weighted routing uses both global and local information in a local path selection procedure. Instead of sending only the identifiers of chosen paths, the NRCC sends the lengths of selected paths along with their identifiers in hybrid weighted routing. Local nodes use transmitted global path length information and local queue length information when they select paths.

In hybrid weighted routing, the NRCC includes many possibly *good* paths by using a larger δ in the path screening procedure. Here, we did not address the problem to find the optimal value of parameter δ . But it is important to find the optimal value because this will remove *very bad* paths at the NRCC and will reduce the network overhead of transmitting NRCC routing decisions. In this study, we set $\delta = \infty$; all alternative paths are selected as candidate paths.

The routing decision process of hybrid routing is as follows. Suppose there are two alternate paths A and B from node i to node j , with globally computed packet delay C_A and C_B , respectively. When node i receives a packet destined to node j , it estimates the delay d_{l_A} and d_{l_B} of links l_A and l_B . Links l_A and l_B are the starting links of path A and B incident at node i . Using the global path delay and the locally estimated link delay, node i selects paths by using the following rule

$$\text{if } k \cdot C_A + (1 - k) \cdot d_{l_A} < k \cdot C_B + (1 - k) \cdot d_{l_B} \text{ then choose path } A$$

$$\text{if } k \cdot C_A + (1 - k) \cdot d_{l_A} > k \cdot C_B + (1 - k) \cdot d_{l_B} \text{ then choose path } B$$

where the parameter k ($0 \leq k \leq 1$) is the weight given to global information. Because this

routing method controls relative importance of global and local information by the weight k , it is called *hybrid weighted* routing. Hybrid weighted routing degenerates to shortest queue routing if k is set to 0 and becomes NRCC routing when k is 1. In section 4, we will show how the performance of hybrid weighted routing changes as different values are assigned to the parameter k .

3.4. Motivation of Packet Routing Performance Study

Since Rudin's seminal performance study [Rudi76], there have been only limited comparisons of routing techniques. General, and extensive comparisons between hybrid routing methods and other non-hybrid routing methods or comparisons among hybrid routing methods have not been done. Hence, one of our major goals has been a comparison of a wide variety of routing algorithms in packet switching networks: random, NRCC, shortest queue, delta, and priority queue routing. We proposed a new hybrid routing method called hybrid weighted routing. In addition to the five existing routing algorithms, we include hybrid weighted routing in this performance study. Especially, we will compare the performance of hybrid weighted routing and the other two hybrid routing techniques. The main issues of our performance study on packet routing algorithms are

- comparison of adaptive and non-adaptive (random, NRCC) routing methods
- comparison of hybrid (delta, priority queue, hybrid weighted) and the other distributed (random, shortest queue) or centralized (NRCC) routing methods
- Comparison of our proposed hybrid weighted routing and the other two hybrid routing methods.

This performance study is done on packet switching networks. We will extend the packet routing performance study to an integrated switching network later. The performance investigation in the packet switching networks will serve as a base case for the further investigation in an hybrid switching network. Also, we will eliminate some non-promising routing techniques based on this investigation in packet switching networks to reduce the size of the integrated switching network study.

3.5. Methodology

We used a discrete event computer simulation method for our performance studies. SIMPAS [Brya81], developed at the University of Wisconsin at Madison, was used in our study. SIMPAS is a discrete, event-oriented system simulation language based on PASCAL. SIMPAS provides a simulation statistics collection facility as well as event generation and event scheduling facilities.

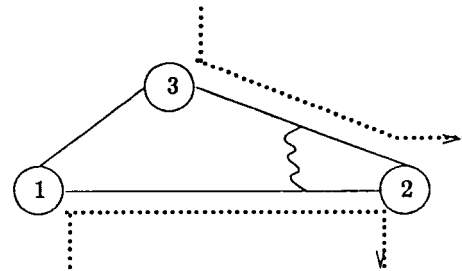
We used regeneration to calculate the averages and variations of network measurements. A simulation is partitioned into several batches. The averages of network performance measures are calculated for each batch and these batch averages are used to calculate the final averages and variations of the performance measures. Each batch consists of a fixed number (batch size) of certain event occurrences. In our packet network simulation, the event is the completion of a packet delivery to its destination. For each simulation, we determined the number of batches and the batch size. After the completion of each batch, we collected the statistics of the batch and continued the simulation. A simulation is continued until the predetermined number of batches are generated.

3.5.1. Four Simulated Networks

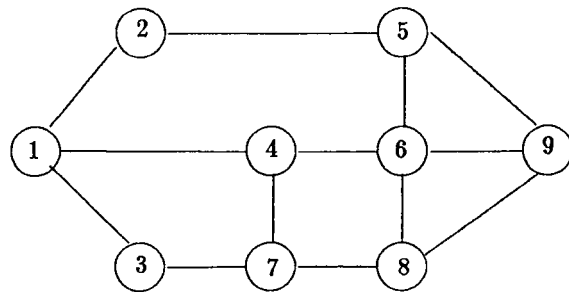
We performed a simulation study with two small networks and two medium size networks. These networks are shown in Figure 3.1. The two small networks were used in Rudin's simulation study [Rudi76]. Small networks usually permit more detailed and cost effective analysis of simulation results. Also, we can validate the correctness of our simulation model by comparing the simulation results to those of Rudin. The other two networks are a nine node ring network with redundant chords between nodes two hops apart and a nine node irregular network. These networks provide some basis for predicting the performance of routing methods in bigger networks.

Network A has four nodes with node 2 as a NRCC. All traffic of network A flows from left to right and the traffic from node 1 to node 2 has two alternative paths. Traffic from node 3 and 4 to node 2 has only one direct path. In network B, traffic from node 1 to node 2 is either routed directly or routed through node 3. All traffic from node 3 to node 2 is routed directly.

Network C is a nine node ring network with redundant chords. The network configuration of network C is similar to that of NASA's proposed space station network [Murr84]. The nodes of network C are connected by full duplex lines. The redundant chords of network C increase the transmission capacity of the network. They allow faster packet deliveries because they connect nodes two hops apart. Additionally, the redundant chords enhance the reliability of the network. In network C, we adopted the convention that all traffic between adjacent nodes (nodes directly connected by backbone ring links) is routed over the connecting backbone links. Other traffic between non-adjacent nodes have two or three alternative first outbound links depending how these nodes are separated each other.



Network B



Network D

For example, there are two alternate paths from node 1 to node 3. One path consists of links (1,2) and (2,3). Alternatively, a packet can be routed directly over a redundant chord (1,3). From node 1, packets destined to node 4 can either be routed to node 2 or be routed to node 3. Packets destined to node 5 can be routed in three different ways. The first alternative path routes packets to node 2. Another routes packets to node 3 through link (1,3). Still another path sends packets to node 8. These examples can be generalized by taking other nodes as source nodes and completely explain alternative paths in network C. Figure 3.2. illustrates the alternative first outbound links in network C. The NRCC of the network is node 1.

Network D is a nine node irregular network. Nodes of network D are connected by full duplex lines. A node pair of network D can have at most two alternative first outbound links. These alternative outbound links are chosen such that the network does not have *loops*. Figure 3.2. shows the alternative paths. The NRCC of the network is located at node 4.

3.5.2. Routing Decision Procedure

The centralized and hybrid routing techniques use the NRCC for global path selections. The NRCC updates routing tables either periodically or asynchronously. In our study, all global routing updates are done periodically. At every routing update interval, local nodes compute the instantaneous delay of each incident link. Let q_{ij} and r_{ij} be the queue length and the transmission capacity of a link (i, j). The instantaneous delay of link (i, j) is

$$d_{ij} = \frac{q_{ij}}{r_{ij}}.$$

Because all links in the simulation network have the same transmission capacity, we use the

		Destination node								
Present node		1	2	3	4	5	6	7	8	9
	1		2	2,3	2,3	2,3,8	3,8,9	8,9	8,9	9
	2	1		3	3,4	3,4	3,4,9	1,4,9	1,9	1,9
	3	1,2	2		4	4,5	4,5	1,4,5	1,2,5	1,2
	4	2,3	2,3	3		5	5,6	5,6	2,5,6	2,3,6
	5	3,4,7	3,4	3,4	4		6	6,7	6,7	3,6,7
	6	4,7,8	4,5,8	4,5	4,5	5		7	7,8	7,8
	7	8,9	5,8,9	5,6,9	5,6	5,6	6		8	8,9
	8	1,9	1,9	1,6,9	1,6,7	6,7	6,7	7		9
	9	1	1,2	1,2	1,2,7	1,7,8	7,8	7,8	8	

(a) Network C

		Destination node								
Present node		1	2	3	4	5	6	7	8	9
	1		2	3	4	2	4	3,4	3,4	2,4
	2	1		3	1,3	5	1,5	1,3	1,3	5
	3	1	2		1,7	2	1,2	7	7	2,7
	4	1	1	1,7		1,6	6	7	6,7	6,7
	5	2,6	2	2	2,6		6	6,9	6,9	9
	6	4	4,5	4,5	4	5		4,8	8	9
	7	3,4	3,4	3	4	4,8	4,8		8	8
	8	6,7	7,9	7	6,7	6,9	6	7		9
	9	5,6	5	5,8	6,8	5	6	6,8	8	

(b) Network D

Figure 3.2. Alternative first links of network C and D.

queue length q_{ij} instead of the delay d_{ij} . With the present queue length and the previous average queue length, Node i computes the new average queue length. The new average delay \bar{q}_{ij} is

$$\bar{q}_{ij} = \alpha \cdot q_{ij} + (1 - \alpha) \cdot \bar{q}_{ij}^{old} \quad \text{where } 0 \leq \alpha \leq 1.$$

Local nodes send the average queue lengths of its incident links to the NRCC. The average queue lengths are used instead of the present queue lengths because the average values allows more stable routing paths.

With the reported average queue lengths, the NRCC selects the most promising paths for each source–destination pair. Usually, NRCC routing chooses a single shortest path for each source–destination pair by solving the *shortest path* problem. In hybrid routing techniques, the NRCC selects multiple alternative paths with different first links for each pair. Rudin [Rudi76] observed that it was necessary only to consider several predetermined *promising* paths instead of considering all possible paths. In his simulation study [Rudi76], he chose the paths by comparing the costs of the predetermined candidate paths. We used the Rudin's approach. In networks A and B, only the traffic from node 1 to 2 has alternative paths. In network C and D, we select paths by comparing the average queue lengths of the candidate paths shown in Figure 3.2. The packet routing in this simulation is loop-free because the all candidate paths in Figure 3.2 are also loop-free.

NRCC routing selects one least cost path for each source–destination pair. In delta, and priority queue routing, the NRCC selects several *good* paths for each pair and sends the selected paths to other nodes as routing tables. In hybrid weighted routing, the NRCC sends the costs of candidate paths as routing tables.

4. PERFORMANCE OF PACKET ROUTING ALGORITHMS

4.1. Introduction

In section 3, we explained the motivation of packet routing performance study. Also, the general configuration of simulated networks was discussed in that section. Here, we present the detailed descriptions of the simulation workloads on each network explained in section 3. These descriptions are summarized in three tables. With the four simulation networks, we compared the performance of six packet routing algorithms: random, shortest queue, NRCC, delta, priority queue, and hybrid weighted routing. The hybrid routing algorithms provides better performance in all simulation networks. Especially, hybrid weighted routing shows the best performance.

4.2. Simulation Traffic

Network A (Figure 3.1) has the same simulation traffic as Rudin's simulation [Rudi76]. The network has balanced traffic from node 1 to node 2 and a surge traffic session that synchronously switches between node pairs (3, 2) and (4, 2). The alternating surge traffic session generates packets on one node pair for 10 000 time units and switches to the other node pair. Only the balanced traffic has alternative routing paths. Also, we measured the average delay of the balanced traffic only. The balanced traffic consists of multiple data sessions each of which generates packets by a Poisson process. The session itself is generated by a Poisson process. The average duration time of the the balanced traffic session is 10 000 time units. On average, there are ten balanced traffic sessions at one time. The total, balanced traffic workload is equivalent to 30 percent channel capacity of links (1, 3) and (1, 4). The time unit

Table 4.1. Simulation Assumptions of Network A and Network B.

Data packet size	Uniformly distributed over [5, 45].
Link transmission capacity	1 character per time unit.
Packet processing time	1 time unit.
Control packet size	25 characters.
Routing update delay	100 time units.
Session duration time of a balanced traffic session	Uniformly distributed over a range [2 000, 18 000].
Average number of active balanced traffic sessions	10.
Regeneration	
Number of regenerations	150.
Number of observations in one regeneration	1 000.
<i>Simulation Workloads</i>	
Network A	Total balanced traffic workload is equivalent to 30 percent channel capacity of links (1, 3) and (1, 4). Surge traffic workload is equivalent to 30 percent channel capacity of links (3, 2) and (4, 2).
Network B	Total balanced traffic workload is equivalent to 30 percent channel capacity of links (1, 2) and (1, 3). Surge traffic workload is equivalent to 30 percent channel capacity of links (1, 2) and (3, 2).

of the system is determined by the relation between the packet length and the transmission link data rate. A trunk in network A takes p time units to transmit a packet of length p . A packet requires 1 time unit of packet processing at a switching node. The length of the control packets carrying global routing information is fixed at 25 characters. The simulation assumptions of network A are summarized in Table 4.1.

We assume that the NRCC prepares routing tables without any processing delay. However, the control packets from local nodes carrying global routing information are not available to the NRCC immediately. These control packets are actually generated at local nodes

and transmitted to the NRCC. The control packets have a higher priority than data packets. At every routing update interval, local nodes send the control packets to the NRCC. To simplify the simulation, we assume that the NRCC receives all control packets after 50 time units. This delay is the transmission time of a packet of length 25 from node 1 to node 2. The control packets carrying routing tables are not actually transmitted because all traffic flows from left to right in network A. We assume that all local nodes receive the routing tables after 50 time units regardless of the distances of local nodes from the NRCC. In real networks, the nodes adjacent to the NRCC receive the routing tables earlier than other nodes. The asynchronous arrivals of routing tables could cause inconsistency problems such as *looping*. A local node receives a new routing table 100 time units after it reported routing information to the NRCC.

Network B (Figure 3.1) also has the measured balanced traffic and the surge traffic which alternates node pairs (1, 2) and (3, 2) synchronously. The characteristics of session arrival and packet generation processes are same as network A. The surge traffic loaded on node pair (1, 2) is routed over link (1, 2) only while the measured traffic on the same node pair is routed over either link (1, 3) or link (1, 2). The simulation assumptions of network B are illustrated in Table 4.1.

Each node pair of network C (Figure 3.1) has a session of balanced traffic. Network C has 72 node pairs. The balanced traffic sessions generate packets by a Poisson process. The packet generation rates of all balanced traffic sessions are the same. In network C, all packets have the same size. One time unit is required to transmit a packet over a link. The packet processing delay at intermediate nodes is 0.05 time unit. Interfering traffic consists of several surge traffic sessions. The source and destination nodes of a surge traffic session are

Table 4.2. Simulation assumptions of network C.

Packet size (data and control)	Constant, one time unit.
Link transmission capacity	1 packet per time unit.
Packet processing time	0.05 time unit.
Global routing update delay	5 time units.
Regeneration	
Number of regenerations	50.
Observations/Regeneration	2 000.
Balanced traffic	A balanced traffic session is loaded to each node pair (total 72 sessions). The packet generation rate of the balanced traffic is varied to simulate different traffic workloads. The rate changes from $\lambda_b = 0.0965$ to $\lambda_b = 0.2325$. The total workload of the 72 balanced traffic sessions at $\lambda_b = 0.0965$ is 6.948 packets/timeunit.
Surge traffic	
Session duration	Uniformly distributed over [100, 300].
Packet generation rate of a session	Uniformly distributed over [0.1, 0.24].
	There are 20 active surge traffic sessions on average. The total surge traffic workload is 3.4 packets/timeunit.

randomly chosen when the session is created. There are on average of 20 surge traffic sessions in the network at one time. Each surge traffic session simulates dynamically varying traffic with short average duration time of 200 time units. Surge traffic sessions are generated by a Poisson process. Also, each interfering surge traffic session generates packets by a Poisson process. We varied the packet generation rates of the balanced traffic sessions to investigate the performance of routing algorithms as a function of a traffic workload. Because we fix the workload of the surge traffic, the surge traffic consists of from 17 percent to 33 percent of the total traffic (excluding control traffic). The varying balanced traffic workloads and the network throughput are illustrated in Table 4.2. The NRCC of network C is node 1. The routing table update delay is 5 time units. This routing update delay only considers the transmission delay of control packets. Networks A and B do not simulate the

control packets carrying routing tables. In networks C and D, both global information packets and routing table packets are actually generated and transmitted. As in network A and B, all arrivals of control packets are synchronized.

Network D (Figure 3.1) is a nine node irregular network. The traffic characteristics of network D is similar to those of network C. A session of balanced traffic is loaded on each node pair. A surge traffic session generator creates surge traffic sessions by a Poisson process. Also, each surge traffic session generates packets by a Poisson arrival process. The source and destination nodes of the surge traffic sessions are determined randomly. The total simulation traffic uses 53 percent of link transmission capacity when random routing is used. The NRCC of network D is node 4 which is the center of the network. Table 4.3 illustrates the simulation assumptions of network D.

Table 4.3. Simulation assumptions of network D.

Packet size (data and control)	Constant, one time unit.
Link transmission capacity	1 packet per time unit.
Packet processing time	0.05 time unit.
Global routing update delay	5 time units.
Regeneration	
Number of regenerations	100.
Observations/Regeneration	2 000.
Balanced traffic	A balanced traffic session is loaded to each node pair (total 72 sessions).
Surge traffic	
Session duration	Uniformly distributed over [100, 300].
Packet generation rate of a session	Uniformly distributed over [0.04, 0.1].
	There are 48 active surge sessions on average.

4.3. Simulation Results

Figure 4.1 shows the average packet delay in network A as a function of the parameter δ of delta routing changes. The performance of delta routing does not change greatly over a wide range of delta values. At $T = 2000$, the network performance is not significantly different within the interval between $\delta=2.0$ and $\delta=10.0$. However, the average delay at $\delta=6.0$ is significantly lower than the average delay at $\delta=1.0$ and the average delay at $\delta \geq 20.0$. We obtain the lowest average delay with $\delta = 3.0$ and 6.0 at update intervals $T = 1000$, and 2000 respectively. As the update interval increases, the optimal performance of delta routing requires larger δ . Intuitively, it is better to rely more on instantaneous local information by using larger δ as global information loses accuracy at larger update intervals.

Note the performance of delta routing deteriorates when the parameter δ takes on values too small or too large. When $\delta = 1.0$, the NRCC chooses paths too selectively and discards many *potentially good* paths. At large δ , the NRCC includes *bad* paths and these bad paths can be selected at local nodes. In both cases, the performance of delta routing decreases. Also, note that the performance difference of delta routing at two routing update intervals, $T = 1000$ and 2000 , changes as different values are assigned to δ . More routing decision responsibility is shifted to the NRCC when smaller δ is used in delta routing. Delta routing has more NRCC routing properties at the smaller δ , and it is better to update global routing more frequently as long as the benefit of more accurate global path selections is not offset by the overhead of the more frequent global routing updates. However, delta routing becomes more like shortest queue routing at larger δ and the frequent global routing updates only increase the control packet transmission overhead with little benefit of more accurate global information. In Figure 4.1, the performance advantage of more frequent ($T = 1000$)

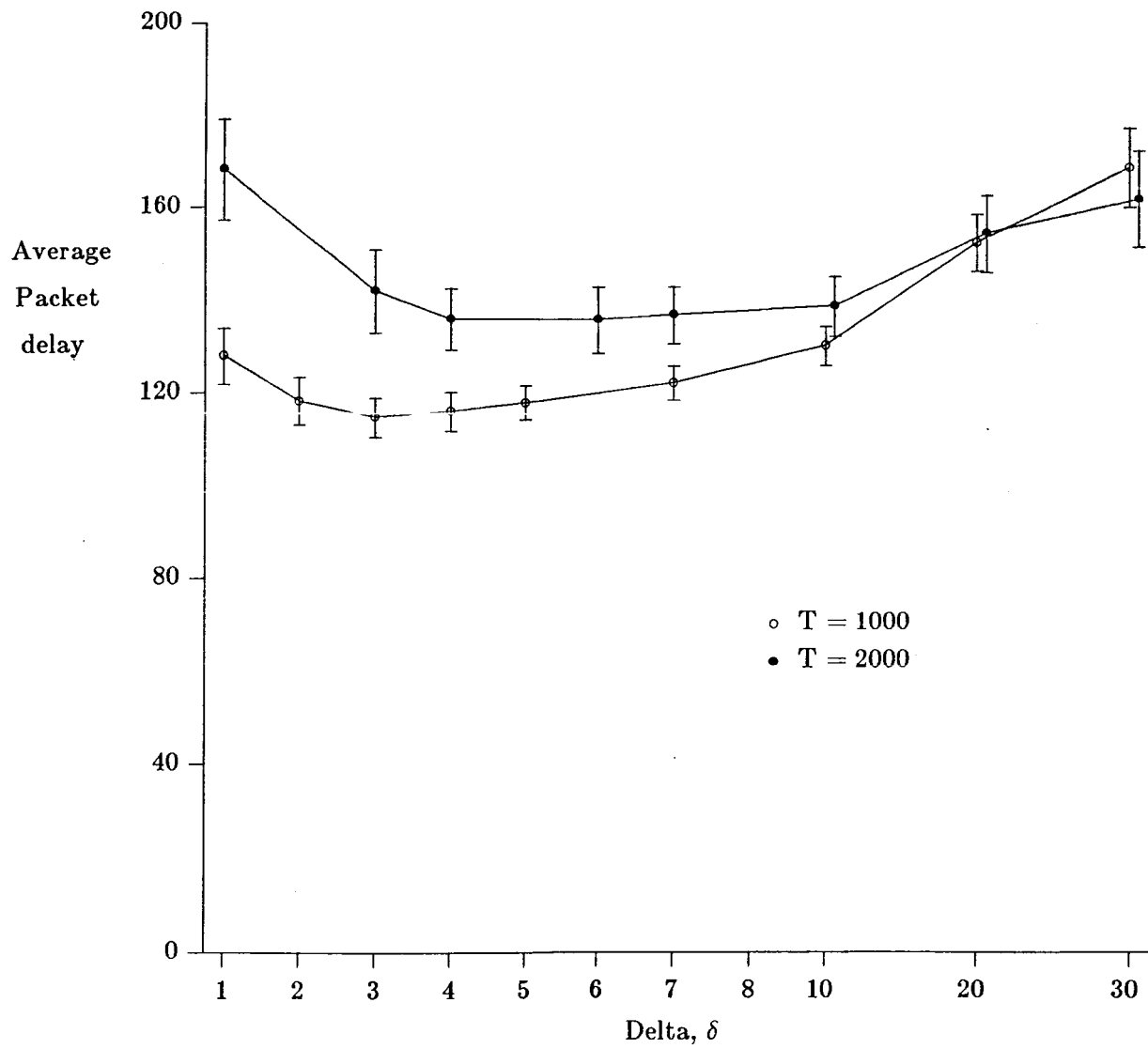


Figure 4.1. Network A.
Effects of delta on the average packet delay in delta routing.

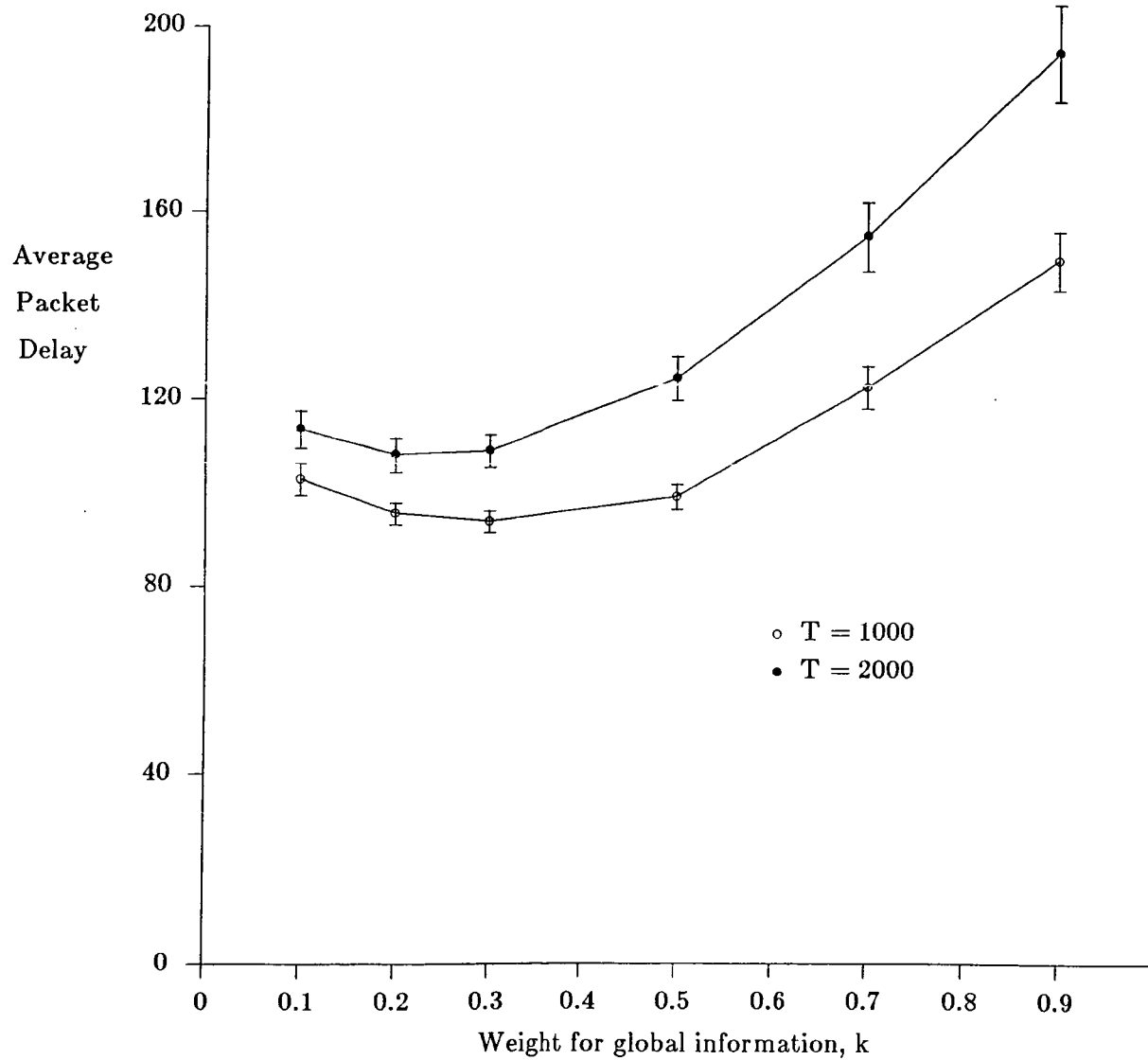


Figure 4.2. Network A.
Effects of parameter k on the average packet delay in hybrid weighted routing.

global updates decreases at larger δ . Two curves meet at $\delta = 20$ and the performance of $T = 2000$ eventually becomes better than the performance of $T = 1000$ at $\delta = 30$.

The average delay of hybrid weighted routing in network A is shown in Figure 4.2 as a function of the parameter k . As with delta routing, the performance of hybrid weighted routing is insensitive to the variations of parameter k . The lowest average delay is obtained with $k = 0.3$, and 0.2 at update intervals $T = 1000$, and 2000 correspondingly. A larger update interval requires less weight (smaller k , less importance) for global routing information.

The performance of the six routing algorithms on network A are compared in Figure 4.3. The average delay of each algorithm is plotted as a function of the update interval T . Random routing and shortest queue routing do not use global routing information, and the average delay of these two algorithms is invariant over all routing update intervals. In this simulation, shortest queue routing provides shorter average delay than random routing, but the differences are not statistically significant. The other routing algorithms which use global information in routing decisions show one common pattern; the optimal performance is achieved at update interval $T = 500$. At a shorter update interval, the network is overloaded with the higher priority control packets. Apparently, the benefit of using more accurate global information does not compensate the transmission overhead of collecting the global information. At larger update intervals, global information becomes so obsolete that it does not reflect network status correctly.

Another fact that we can observe from Figure 4.3 is that the performance of NRCC routing deteriorates more quickly than the three hybrid routing methods as the global routing update interval increases from the optimal update interval. The hybrid routing methods

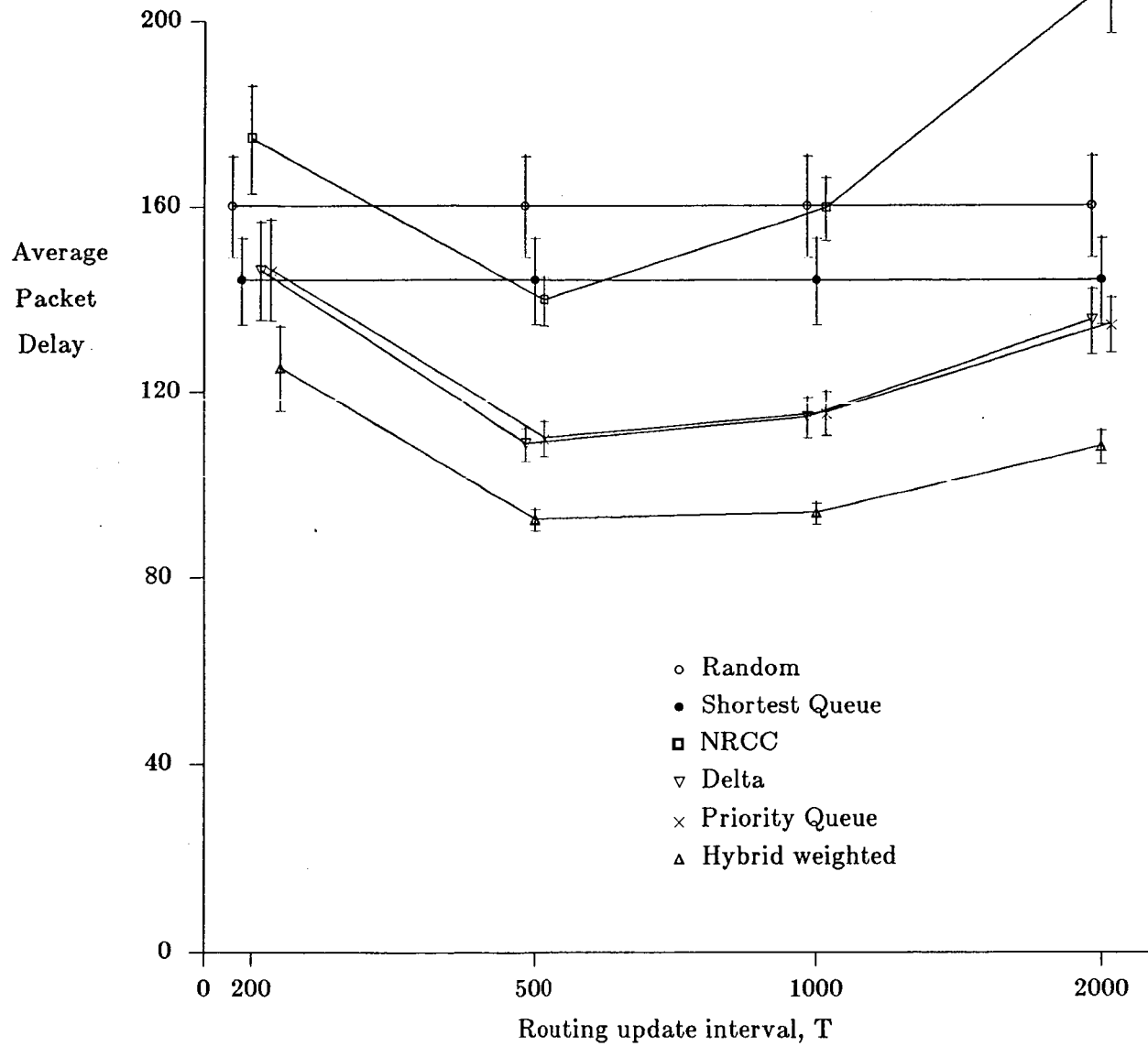


Figure 4.3. Network A.
Performance of routing algorithms as a function of T.

use both global and local information and their performance shows more robustness against the aging global information than NRCC routing which uses only global information. Let us compare the performance of NRCC routing with the performance of random and shortest queue routing to see the sensitivity of NRCC routing to the routing update interval. NRCC routing is slightly better than both random and shortest queue routing when $T = 500$, but is equal or much worse than these two algorithms at larger intervals. We can see how the value of global information drops as one snapshot of global network configuration is used for longer time.

Delta routing and hybrid weighted routing were simulated with the optimal parameters, those that produced the best routing performance in Figure 4.1 and Figure 4.2. Because Figure 4.1 and 4.2 show that delta routing and hybrid weighted routing are not very sensitive to the parameters δ and k , we did not determine the optimal parameters for priority queue routing. Priority queue routing used the same parameters used in delta routing. As Rudin [Rud76] claimed, delta routing performs significantly better than other non-hybrid routing methods at $T = 500$ and $T = 1000$. The performance of delta routing and priority queue routing is very similar at all simulated routing update intervals. Hybrid weighted routing is the best routing method. The delay of hybrid weighted routing is significantly smaller than all other routing methods at all simulated routing update intervals. It has 14 to 20 percent less average delay than delta routing. The average delay of hybrid weighted routing increases least steeply among the simulated routing algorithms which use global information. The relative performance enhancement by hybrid weighted routing over the other hybrid routing methods and NRCC routing becomes more significant at larger update intervals. From this fact, we can see that hybrid weighted routing is most robust against the aging global information.

The average delay of the six routing algorithms on network B is shown in Figure 4.4. The parameters of delta routing and hybrid weighted routing were optimized for each routing interval. Figure 4.4 shows the superiority of adaptable routing methods. Random routing, an non-adaptable routing method, has significantly larger average delay than shortest queue routing. Also, NRCC routing is worse than shortest queue at all routing update intervals. In network B, a surge traffic session jumps back and forth between node pairs (3, 2) and (1, 2). With shortest queue routing, node 1 can locally detect increasing traffic on link (1, 2) and routes packets over link (1, 3) when the surge traffic is loaded on node pair (1, 2). However, NRCC routing sends packets over link (1, 2) until the next routing update if link (1, 2) was selected at the previous routing update. NRCC routing suffers more from its non-adaptability at larger routing update intervals. NRCC routing is even worse than random routing at $T = 2000$. The hybrid routing algorithms avoid the non-adaptability problem of NRCC routing by using local network information in their routing decisions.

Figure 4.3 and Figure 4.4 show the same performance pattern of NRCC, delta, priority queue, and hybrid weighted routing; the performance improves first and then degrades as the routing interval increases. The best performance of the three algorithms, NRCC, delta, and priority queue routing, is achieved at $T = 500$. However, the best performance of hybrid weighted routing is obtained at $T = 1000$. The performance of hybrid routing does not deteriorate much even at $T = 2000$. The average delay of hybrid weighted routing increases only 4.7 percent at $T = 2000$ from the minimum average delay at $T = 1000$. This result reconfirms both the performance superiority and the robustness of hybrid weighted routing.

Figure 4.5 shows the performance of routing methods in network C. The total workload of the simulated traffic is 15.64 packets/timeunit (excluding the control packets). Delta rout-

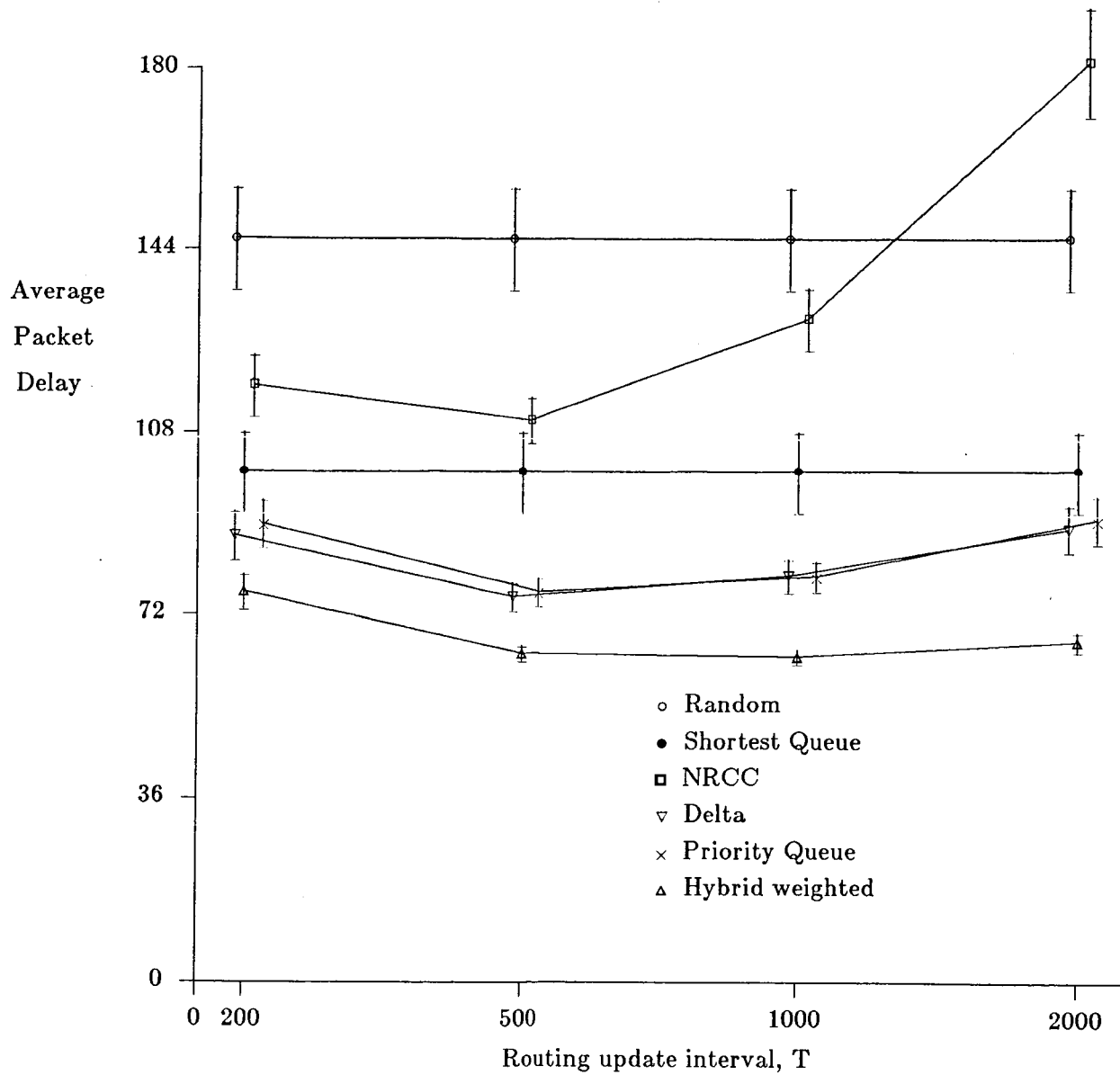


Figure 4.4. Network B.
Performance of routing algorithms as a function of T.

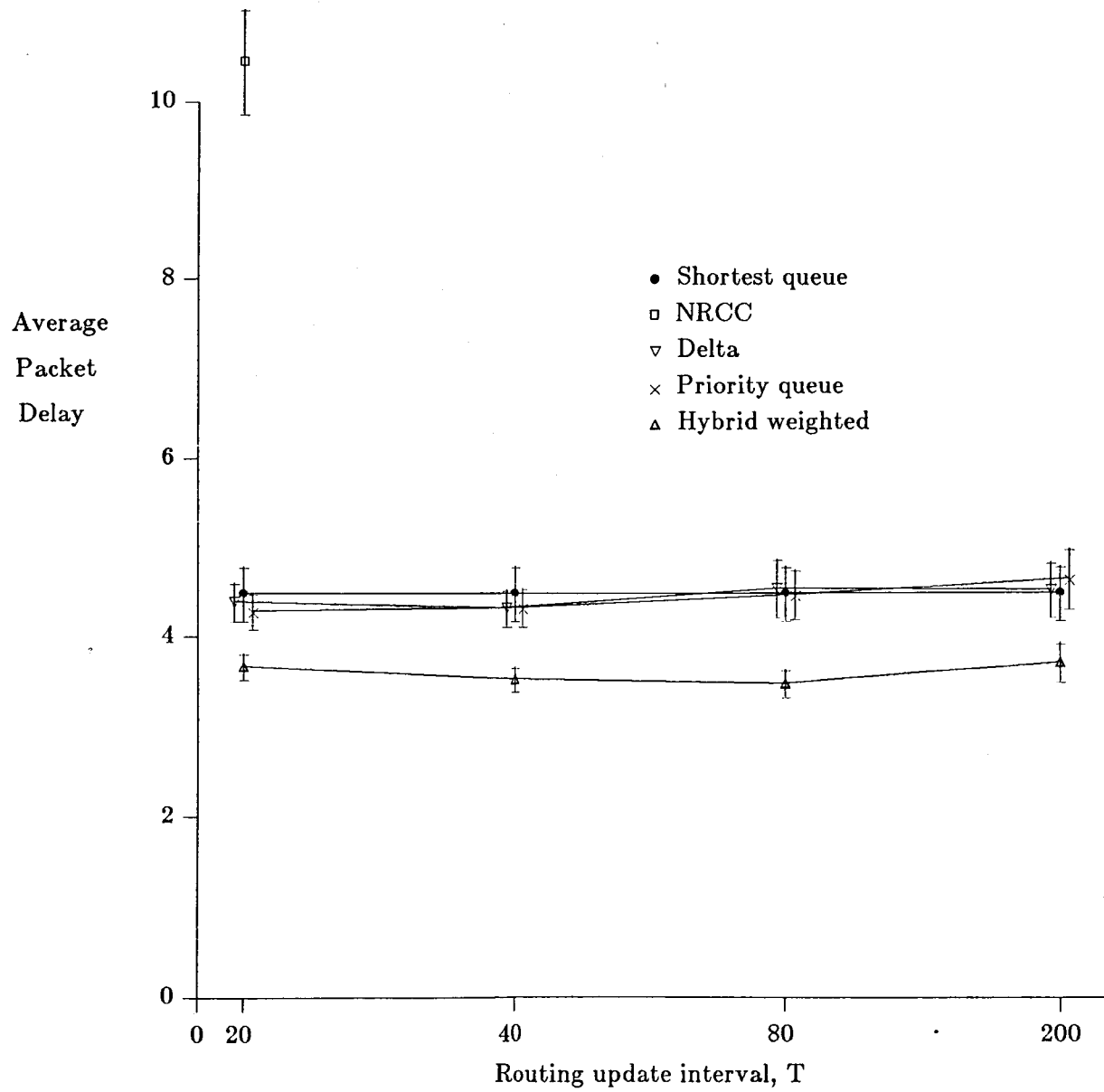


Figure 4.5. Network C.
Performance of routing algorithms as a function of T .

ing and hybrid weighted routing used the parameters which were optimized for each routing update interval. NRCC routing and random routing cannot effectively handle the simulation traffic. Only the average delay of NRCC routing at $T = 20$ can be shown on this figure. The average delay of NRCC routing increases monotonically and rapidly as the routing update interval increases. At larger routing update intervals, old global information did not correctly reflect the rapidly changing network conditions. Because old global information is so unreliable in this simulation, more routing decision responsibility is shifted from the NRCC to local nodes in the two hybrid routing methods, delta and priority queue routing. Hence, the average delay of shortest queue routing, delta routing and priority queue routing is statistically indistinguishable. At $T = 200$, delta routing degenerates to shortest queue routing. Only hybrid weighted routing shows consistently good performance. As in the small network simulations, hybrid weighted routing shows the best performance. Hybrid weighted routing has 16–23 percent shorter average delay than delta routing.

Figure 4.6 shows the performance of the routing methods as a function of simulation workload. Delta routing and hybrid routing used the same optimal parameters used in the previous simulation shown in Figure 4.5. With random routing and NRCC routing, the network saturated at low workloads. Delta routing and shortest queue routing perform almost identically at lower workloads. However, delta routing outperforms shortest queue routing at larger workloads. At low workloads, the probability of any regions of the network being saturated is low, although the network may have unbalanced traffic. At higher workloads, there are more chances that some areas of the network are saturated. Because shortest queue routing only knows local status, it may route packets over *locally good* paths which eventually lead to the overloaded areas. Delta routing avoids the overcrowded areas using global information. Hybrid weighted routing outperforms all other routing methods at all

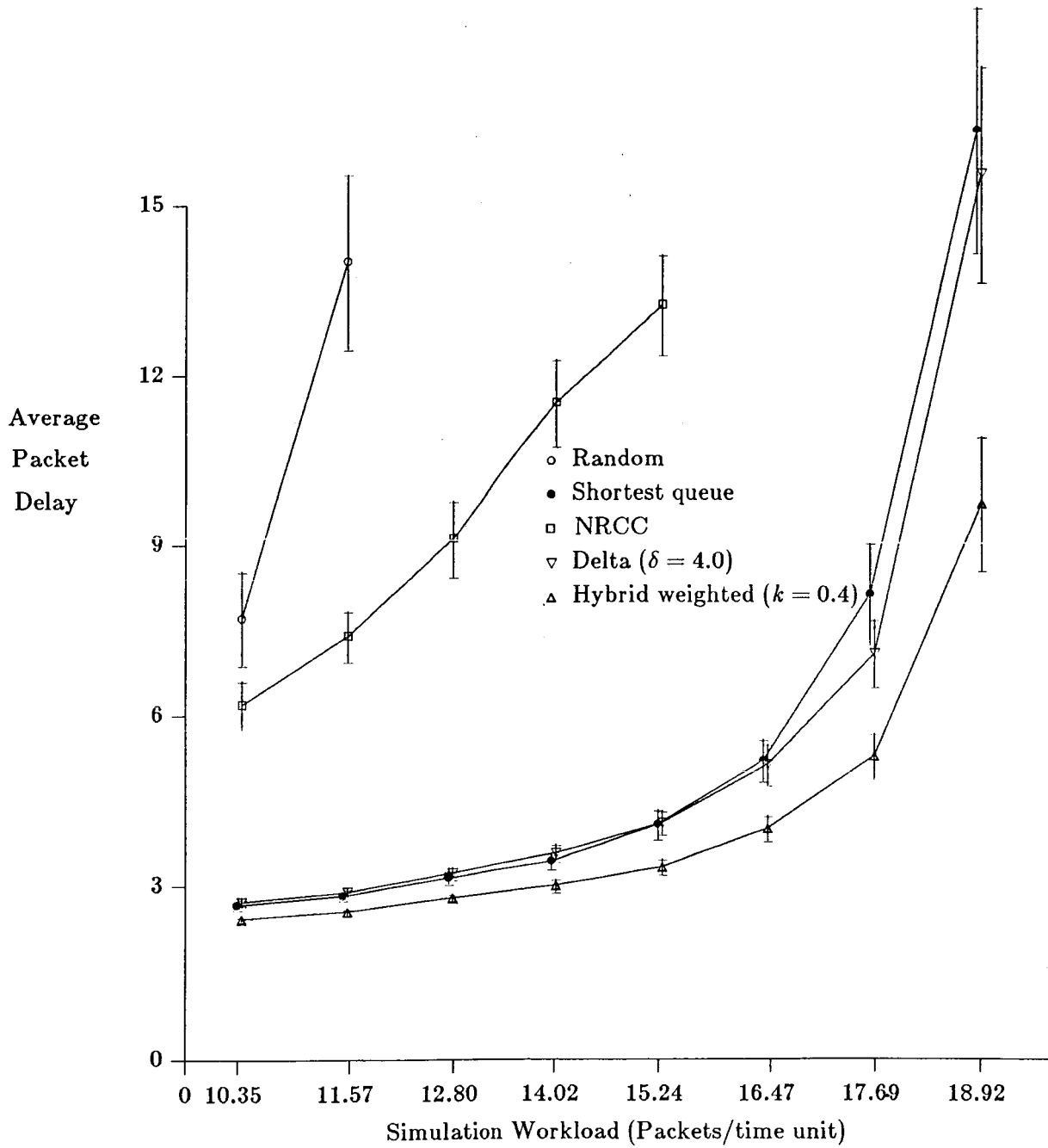


Figure 4.6. Network C (Routing update interval $T = 40$).
Performance of routing algorithms as a function of simulation workload.

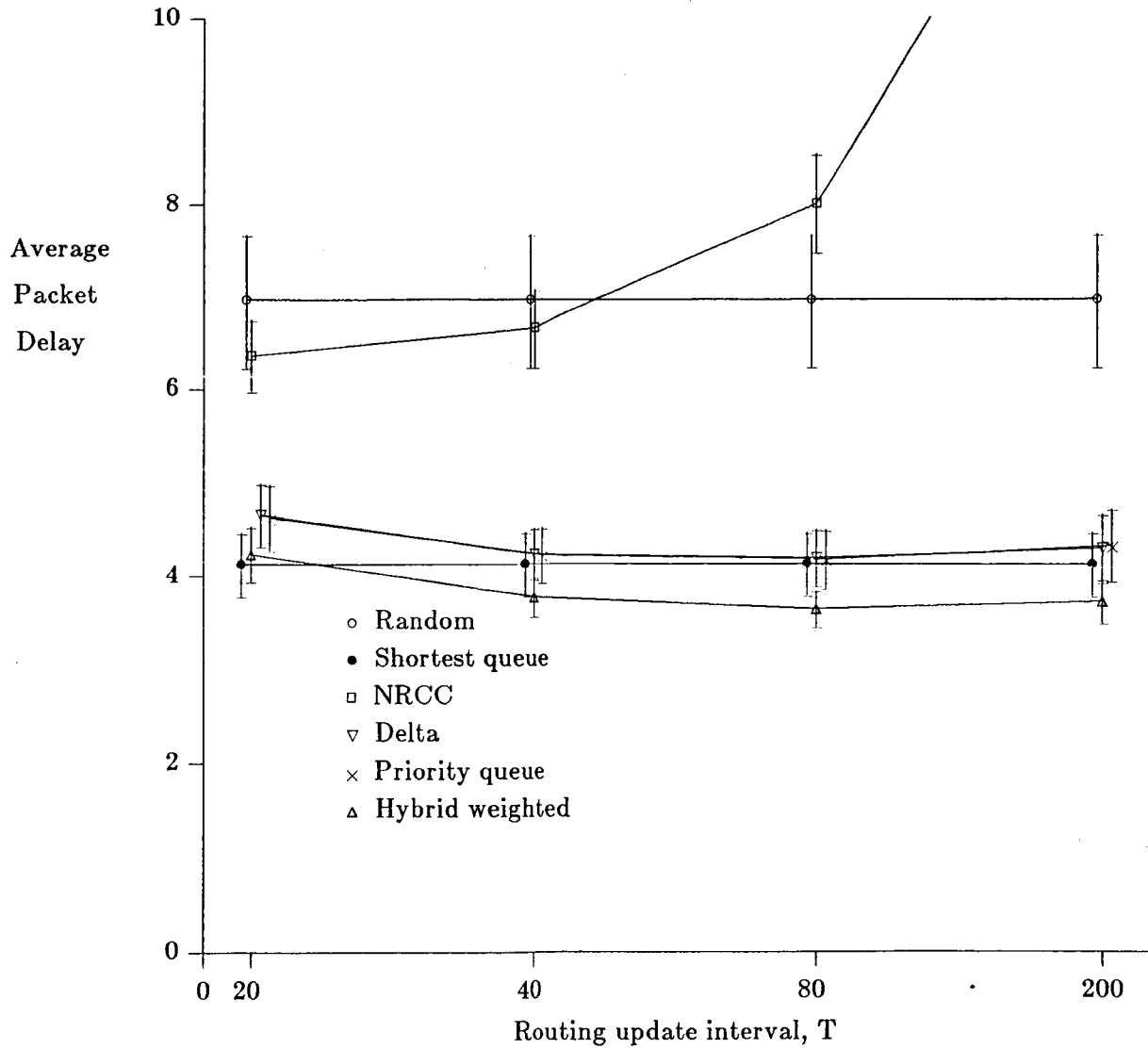


Figure 4.7. Network D.
Performance of routing algorithms.

workloads. The advantage of hybrid weighted routing becomes more significant at larger workloads. Compared to the next best routing method, i.e. delta routing, hybrid weighted routing has 11 percent shorter delay at the lowest workload and 37 percent shorter delay at the highest workload.

The simulation results on network D are shown in Figure 4.7. Again, the performance of NRCC routing deteriorates as routing update interval increases. NRCC routing is even worse than random routing at large update intervals because NRCC routing continuously uses old routing paths derived from obsolete network conditions until new routing tables are created. Shortest queue, priority queue and delta routing have much better performance than random routing and NRCC routing. This shows the advantage of adaptive routing methods in this dynamically changing network condition. Hybrid weighted routing outperforms other methods, but the advantage of hybrid weighted hybrid is not so great as in the earlier simulations. The advantage of hybrid weighted routing is most prominent at $T = 80$.

4.4. Summary

We can draw several conclusions from the above simulation results. First, we summarize the simulation results according to the main goals of this simulation study:

- Comparison of adaptive and non-adaptive routing methods.
- Comparison of hybrid and non-hybrid routing techniques.
- Comparison of hybrid weighted routing and the other routing methods.

In all simulations, adaptive routing outperformed non-adaptive routing methods. This simulation results agreed with former performance studies on packet routing algorithms [FuKl71], [Rudi76], [ChBN81]. Fultz and Kleinrock [Fukl71], and Rudin [Rudi76] showed the

performance advantage of shortest queue routing and delta routing over non-adaptive routing methods such as random or NRCC routing. Also, Chou *et al.* confirmed the superiority of adaptive routing in dynamically changing network environments.

The three hybrid routing methods showed significantly better performance than the other hybrid routing methods in networks A and B. In networks C and D, the performance enhancement of the two hybrid routing methods, delta and priority queue routing, over shortest queue routing is not so great as in networks A and B. Only hybrid routing methods showed consistently better performance. The simulations on network C and D showed the problem of aging global information. Because old global information is so unreliable in these networks, more routing decision responsibility is transferred to local nodes which decides routing paths based on instantaneous local network information. Actually, delta routing degenerated to shortest queue routing at larger routing update intervals. However, a further investigation with network C (shown in Figure 4.6) shows that hybrid routing is significantly better than the other non-hybrid routing including shortest queue routing at higher workloads.

Hybrid weighted routing outperforms the other five simulated algorithms in all simulations. Hybrid weighted routing significantly improves the performance of the two small networks and the nine node ring network. Hybrid routing, which considers both the global path costs and local link costs, handles the network status variations consistently well. Hybrid weighted routing is the best algorithm in network D, but the performance improvement is not so great as in the other three networks. The other two hybrid routing algorithms, delta routing and priority queue routing, are not so outstanding as hybrid weighted routing. The performance of these two hybrid routing algorithms are almost identical.

In addition to the performance superiority, hybrid weighted routing has the most robustness against aging global information. Hybrid weighted routing achieves greater performance enhancements over NRCC, delta, and priority queue routing at larger routing update intervals. Also, Figure 4.6 shows that the relative performance advantage of hybrid weighted routing over other routing is more prominent at higher workloads.

Based on these simulation results, we plan to extend the packet routing performance study to an integrated switching network. In the integrated switching network simulation, we will exclude random routing and priority queue routing. Random routing showed much worse performance than the other routing methods in the simulations on packet switching networks. The performance of delta routing and priority queue routing was very similar in all simulations. We expect that the two methods continuously show similar performance in integrated switching networks.

5. FUTURE WORK

In section 4, we compared the performance of several packet routing methods in packet switching networks. The simulated packet routing methods were random, NRCC, shortest queue, delta, priority queue, and hybrid weighted routing. Based on the results from the packet switching network simulations, we plan to extend our performance study to the integrated switching network with two boundary management schemes: fixed boundary and movable boundary. Before clarifying the each research goals, we first list the required research tools for the further investigations.

First, we need to develop adaptive circuit (call) routing methods. Initially, we will use the simplest call routing method, i.e. random call routing, in the future integrated network simulations. The random call routing is well suited for initial studies because they will concentrate primarily on the switching part of the integrated switching network. To investigate the circuit switching part and to study the interaction between the packet and circuit switching parts, we need to include more advanced call routing methods in future simulations. Several adaptive call routing methods were introduced in section 2. These methods were developed in pure circuit switching environments. Hence, these method do not consider the interaction of circuit traffic and packet traffic of the integrated network. We need to either develop new call routing methods or improve the existing call routing methods.

With these extensions, we will extend our research on the integrated switching network. There are two main future research areas:

- The performance comparison of packet routing methods in the integrated switching network
- The performance of circuit routing algorithms in the integrated switching network

- Implementation of several promising packet and circuit routing algorithms in a real integrated network system to compare the results from the real system and the simulations.

Each of these research goals will be discussed in a greater detail.

First, we will extend the performance study of packet routing algorithms in an integrated network. All earlier integrated network performance studies shown in section 2 [Kumm 74], [Fish 76], [Wein 79], [Wein 80], [Kohn 84], [Will 84], and [Lian 85] focused on the microscopic operation of integrated switching; they dealt with the frame structure in one separated trunk. They evaluated the performance gains obtained from the movable boundary management scheme. No one has evaluated the performance of a whole complete integrated network with multiple nodes and multiple interconnections. We will address the performance of integrated switching networks in real communication system environment. We will investigate how the performance gains of a movable boundary at individual trunks improve the performance of one complete integrated switched network system. An extension of this study is highly correlated with the performance study of packet routing algorithms. We will investigate packet routing algorithms which optimally utilized the additional packet transmission capacity gained by using the flexible boundary policy in integrated switching networks.

There are two types of network state variations in a heterogeneous communication system: the variation of traffic workloads and the variation of network components. Most performance studies of packet switching routing algorithms focused on the adaptability of routing techniques against the traffic workload variations. The performance studies by Fultz and Kleinrock [FuKl71], Rudin [Rudi76], and Chou *et al.* [ChBN81] dealt with the adaptability of packet routing techniques against the packet traffic fluctuations. The network component

variations such as link failures and link recoveries are difficult to include in simulation or analytic performance studies because network components change far less dynamically than traffic workloads. As pointed out in section 1, the heterogeneous traffic has another important traffic variation, the variation of packet and circuit switched traffic mix. The variations of the traffic mix in integrated networks result in variations of packet transmission capacity. The total transmission capacity for packet switching including temporary idle circuit switching capacity fluctuates dynamically as the traffic mix changes. Packet routing algorithms of integrated networks must adapt effectively to dynamically changing traffic mix and traffic workloads. We will extend the performance evaluations of packet routing algorithms to integrated switched networks based on the results from the packet switched network simulations shown in section 4. This performance study in integrate networks will discover the effective routing procedures which handle complex network condition variations of integrated networks.

The second future research goal is to investigate the circuit switching part of the integrated network. We argued in section 1 that a movable boundary integrated network could handle the traffic mix variations efficiently by dynamically shifting network resources to different types of traffic. In addition to the dynamic resource sharing, the integrated network can enhance the network performance by routing (or establishing) the packets and calls over the paths with relatively smaller traffic workloads. Packet routing methods, which deal with packets, were extensively investigated in section 4. Also, our first goal is to extend the packet routing algorithm performance study to integrated networks with two boundary management schemes. However, in this packet routing performance analysis, random call routing will be used because the analysis will concentrate on the packet switching part of integrated networks. Random call routing, the simplest routing method, does not adapt to

network state changes. In the future, we will analyze the performance of adaptive call routing methods which adapt to packet/circuit traffic variations. We expect that adaptive call routing methods will improve the performance of the integrated switching network.

The third future research goal is to implement several promising packet and circuit routing algorithms in a real integrated switching network to compare the performance of routing methods from real system and from simulations. We originally planned to implement these routing algorithms in the NASA's space station network [Murr84]. However, the hardware implementation of the space station network has not yet been completed. Alternatively, we may implement the routing methods in a hypercube machine. A hypercube machine actually generates and passes messages.

REFERENCES

- [ArSt79] E. Arthurs, and B. W. Stuck, "A Theoretical Traffic Performance Analysis of an Integrated Voice-Data Virtual Circuit Packet Switch", *IEEE Transactions on Communications*, Vol. COM-27, No. 7, July 1979, pp. 1104-1111.
- [BeGG84] D. P. Bertsekas, E. L. Gafni, and R. G. Gallager, "Second Derivative Algorithm for Minimum Delay Distributed Routing in Networks", *IEEE Transactions on Communications*, Vol. COM-32, No. 8, August 1984, pp. 911-919.
- [BoFR77] R. R. Boorstyn, and H. Frank, "Large-Scale Network Topological Optimization", *IEEE Transactions on Communications*, Vol. COM-25, No. 1, January 1977, pp. 29-47.
- [BoLi81] R. R. Boorstyn, and A. Livne, "A Technique for Adaptive Routing in Networks", *IEEE Transactions on Communications*, Vol. COM-29, No. 4, April 1981, pp. 474-480.
- [Brya81] R. M. Bryant, "SIMPAS 5.0 User Manual", *Computer Sciences Department, University of Wisconsin at Madison, Technical Report #456*, November 1981.
- [ChMe83] M. S. Chen, and J. S. Meditch, "A Distributed Adaptive Routing Algorithm", *Proceedings of the International Conference on Communications*, Boston, MA., June 1983, pp. 484-492.
- [ChBN81] W. Chou, A. W. Bragg, and A. A. Nilsson, "The Need for Adaptive Routing in the Chaotic and Unbalanced Traffic Environment", *IEEE Transactions on Communications*, Vol. COM-29, No. 4, April 1981, pp. 481-490.
- [CoVe75] G. J. Coviello, and P. A. Vena, "Integration of Circuit/Packet Switching by a SENET (Slotted Envelope Network) Concept", *Proceedings of the National Telecommunications Conference*, 1975, pp. (42-12)-(42-17).
- [DDRP76] A. Danet, R. Despres, A. Le Rest, G. Pichon, and S. Ritzenthaler, "The French Public Packet Switching Service: The TRANSPAC Network", *Proceedings of the International Conference on Computer Communications*, Toronto, Ont., Canada, August 1976, pp. 251-260.
- [FiHa76] M. J. Fischer, and T. C. Harris, "A Model for Evaluating the Performance of an Integrated Circuit-and Packet-Switched Multiplex Structure", *IEEE Transactions on Communications*, Vol. COM-24, No. 2, February 1976, pp. 195-202.
- [FrCh71] H. Frank, and W. Chou, "Routing in Computer Networks", *Networks*, Vol. 1, 1971, pp. 99-112.
- [FrCh72] H. Frank, and W. Chou, "Topological Optimization of Computer Networks", *Proceedings of the IEEE*, Vol. 60, No. 11, November 1972, pp. 1385-1397.

- [FrGK73] L. Fratta, M. Gerla, and L. Kleinrock, "The Flow Deviation Method," *Networks*, Vol. 3, No. 3, 1973, pp. 97-133.
- [FuKl71] G. L. Fultz, and L. Kleinrock, "Adaptive Routing Techniques for Store-and-Forward Computer-Communication Networks", *Proceedings of the International Conference on Communications*, Montreal, P.Q., Canada, June 1971, pp. (39-1)-(39-8).
- [Gall77] R. G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation", *IEEE Transactions on Communication*, Vol. COM-25, No. 1, January 1977, pp. 73-85.
- [GiHu83] A. Girard, and S. Hurtubise, "Dynamic Routing and Call Repacking in Circuit-Switched Networks", *IEEE Transactions on Communications*, Vol. COM-31, No. 12, December 1983, pp. 1290-1294.
- [GiFr78] I. Gitman, and H. Frank, "Economic Analysis of Integrated Voice and Data Networks: A Case Study", *Proceedings of the IEEE*, Vol. 66, No. 11, November 1978, pp. 1549-1570.
- [GiHO81] I. Gitman, W. Hsieh, and B. J. Occhiogrosso, "Analysis and Design of Hybrid Switching Networks", *IEEE Transactions on Communications*, Vol. COM-29, No. 9, September 1981, pp. 1290-1300.
- [HiKY74] K. Hirota, M. Kato, and Y. Yoshida, "A Design of Packet Switching System", *Proceedings of the International Conference on Computer Communications*, Stockholm, Sweden, August 1974, pp. 151-162.
- [HsGO78] W. Hsieh, I. Gitman, and B. Occhiogrosso, "Design of Hybrid-Switched Networks for Voice and Data", *Proceedings of the International Conference on Communications*, 1978, pp. 20.1.1-20.1.9.
- [IySi83] B. R. Iyer, and J. B. Sinclair, "Analysis of Data Delays for an Integrated Data Voice Link", *International Conference on Communications*, Boston, MA., June 1983, pp. 1349-1353.
- [JeKu76] C. J. Jenny, and K. Kummerle, "Distributed Processing Within an Integrated Circuit/Packet-Switching Node", *IEEE Transactions on Communications*, Vol. COM-24, No. 10, October 1976, pp. 1089-1100.
- [KeKl80] P. Kermani, and L. Kleinrock, "A Tradeoff Study of Switching Systems in Computer Communication Networks", *IEEE Transactions on Communications*, Vol. COM-29, No. 12, December 1980, pp. 1052-1060.
- [Klei64] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*. New York: McGraw Hill, 1964.
- [KoPi84] A. G. Konheim, and R. L. Pickholtz, "Analysis of Integrated Voice/Data

- Multiplexing", *IEEE Transactions on Communications*, Vol. COM-32, No. 2, February 1984, pp. 140-147.
- [Kumm74] K. Kummerle, "Multiplexor Performance for Integrated Line- and Packet-Switched Traffic", *Proceedings of the International Conference on Computer Communications*, Stockholm, Sweden, August 1974, pp. 507-515.
- [Lian85] T. Y. Liang, "A simpler Approach to "Analysis of Integrated Voice/Data Multiplexing"", *IEEE Transactions on Communications*, Vol. COM-33, No. 1, January 1985, pp. 104-106.
- [Lipp83] R. P. Lippman, "Survivable Routing Procedures for Circuit-Switched Satellite-Terrestrial Networks", *International Conference on Communications*, Boston, MA., June 1983, pp. 17-22.
- [MaSc81] B. Maglaris, and M. Schwartz, "Performance Evaluation of a Variable Frame Multiplexer for Integrated Switched Networks", *IEEE Transactions on Communications*, Vol. COM-29, No. 6, June 1981, pp. 800-807.
- [Mati76] J. Martin, *Telecommunications and the Computer*. 2nd Edition. Englewood Cliffs, N.J.: Prentice-Hall, 1976.
- [MaFu81] H. J. Matt, and K. Fussaenger, "Integrated Broad-Band Communication Using Optical Networks-Results of an Experimental Study", *IEEE Transactions on Communications*, Vol. COM-29, No. 6, June 1981, pp. 868-885.
- [MiHT75] H. Miyahara, T. Hasegawa, and Y. Teshigawasa, "A Comparative Evaluation of Switching Methods in Computer Communication Networks", *Proceedings of the International Conference on Communications*, San Francisco, CA., 1975, pp. (6-6)-(6-10).
- [MiTH78] H. Mayahara, Y. Teshigawara, and T. Hasegawa, "Delay and Throughput Evaluation of Switching Methods in Computer Communication Networks", *IEEE Transactions on Communications*, Vol. COM-26, No. 3, March 1978, pp. 337-344.
- [McWa77] J. M. McQuillan, D. C. Walden, "The ARPA Network Design Decisions," *Computer Networks*, Vol. 1, August 1977, pp. 243-289.
- [MoKa81] K. Morino, and T. Kawaoka, "Roles of Network Architecture on ISDN", *Proceedings of the National Telecommunications Conference*, New Orleans, LA., 1981, pp. B2.1.1-B2.1.5.
- [Murr84] N. D. Murray, Personal communication.
- [OGHF77] B. Occhiogrosso, I. Gitman, W. Hsieh, and H. Frank, "Performance Analysis of Integrated Switching Communications Systems", *Proceedings of the National Telecommunications Conference*, 1977, pp. (12:4-1)-(12:4-13).

- [OkMM84] Y. Okano, M. Makino, and T. Miki, "Transmission Network Optimization in Junction Networks", *Proceedings of the International Conference on Communications*, Amsterdam, The Netherlands, May 1984, pp. 185-188.
- [PeRS84] J. -L. Pernin, T. M. Randall, and A. J. Schwartz, "Integrating Subscribers into Evolving Digital Networks", *Proceedings of the International Conference on Communications*, Amsterdam, The Netherlands, May 1984, pp. 204-208.
- [PKRJ76] E. Port, K. Kummerle, H. Rudin, C. Jenny, and R. Zafiropulo, "A Network Architecture for the Integration of Circuit and Packet Switching", *Proceedings of the International Conference on Computer Communications*, Toronto, Ont., Canada, August 1976, pp. 505-513.
- [Reed84] D. A. Reed, "Integrated Communication on a Fiber Optic Network: A Feasibility Study," *Unpublished Note, Dept. Computer Science, University of Illinois at Urbana-Champaign*, 1984.
- [RoTW77] M. J. Ross, A. C. Tabbot, and J. A. Waite, "Design Approaches and Performance Criteria for Integrated Voice/Data Switching", *Proceedings of the IEEE*, Vol. 65, No. 9, September 1977, pp. 1283-1295.
- [Rudi76] H. Rudin, "On Routing and "Delta Routing": A Taxonomy and Performance Comparison of Techniques for Packet-Switched Networks", *IEEE Transactions on Communications*, Vol. COM-24, No. 1, January 1976, pp. 43-59.
- [Rudi78] H. Rudin, "Studies on the Integration of Circuit and Packet Switching", *Proceedings of the International Conference on Communications*, Toronto, Ont., Canada, June 1978, pp. 20.2.1-20.2.7.
- [Schu79] D. Schutzer, "An Analysis of a Distributed Switching Network with Integrated Voice and Data in Support of Command and Control", *IEEE Transactions on Communications*, Vol. COM-27, No. 7, July 1979, pp. 1124-1130.
- [ScCh76] M. Schwartz, and C. K. Cheung, "The Gradient Projection Algorithm for Multiple Routing in Message-Switched Networks," *IEEE Transactions on Communications*, Vol. COM-24, No. 4, April 1976, pp. 449-456.
- [ScSt80] M. Schwartz, and T. Stern, "Routing Techniques Used in Computer Communication Networks", *IEEE Transactions on Communications*, Vol. COM-28, No. 4, April 1980, pp. 539-552.
- [Sega77] A. Segall, "The Modeling of Adaptive Routing in Data-Communication Networks", *IEEE Transactions on Communications*, Vol. COM-25, No. 1, January 1977, pp. 85-95.
- [Sega81] A. Segall, "Advances in Verifiable Fail-Safe Routing Procedures", *IEEE Transactions on Communications*, Vol. COM-29, No. 4, April 1981, pp. 491-497.

- [ShTa84] N. Shimasaki, and Y. Takeshige, "A Study on Multiple Mode Business Information Service Networks", *Proceedings of the International Conference on Communications*, 1984, pp. 209-214.
- [StMo83] L. Stolzenberg, and R. Morlinger, "Private Information Network for Data, Text and Voice Communications", *Proceedings of the International Conference on Communications*, Boston, MA., June 1983, pp. 837-841.
- [Tane81] A. S. Tanenbaum, *Computer Networks*. Englewood Cliffs, N. J.: Prentice-Hall, 1981.
- [WeMF79] C. J. Weinstein, M. L. Malpass, and M. J. Fischer, "Data Traffic Performance of an Integrated Circuit- and Packet-Switched Multiplex Structure", *International Conference on Communications*, Boston, MA., June 1979, pp. 24.3.1-24.3.5.
- [WeMF80] C. J. Weinstein, M. L. Malpass, and M. J. Fischer, "Data Traffic Performance of an Integrated Circuit- and Packet-Switched Multiplex Structure", *IEEE Transactions on Communications*, Vol. COM-28, No. 6, June 1980, pp. 873-878.
- [WiLe84] G. F. Williams, and A. Leon-Garcia, "Performance Analysis of Integrated Voice and Data Hybrid-Switched Links", *IEEE Transactions on Communications*, Vol. COM-32, No. 6, June 1984, pp. 695-706.
- [Yum81] T. P. Yum, "The Design and Analysis of Semidynamic Deterministic Routing Rule", *IEEE Transactions on Communications*, Vol. COM-29, No. 4, April 1981, pp. 498-504.
- [YuSc81] T. P. Yum, and M. Schwartz, "The Join-Biased-Queue Rule and Its Application to Routing in Computer Communication Networks", *IEEE Transactions on Communications*, Vol. COM-29, No. 4, April 1981, pp. 505-511.
- [Zafi74] P. Zafiropulo, "Flexible Multiplexing for Networks Supporting Line Switched and Packet Switched Data Traffic", *Proceedings of the International Conference on Computer Communications*, Stockholm, Swede, August 1974, pp. 517-523.
- [ZaPK76] P. Zafiropulo, E. Port, and K. Kummerle, "Extension of Circuit-Switched User/Network Interface to Packet-Switching", *Proceedings of the International Conference on Computer Communications*, Toronto, Ont., Canada, August 1976, pp. 515-522.